11:29:49 OCA PAD INITIATION - PROJECT HEADER INFORMATION 10/07/87

Active

Project [redacted]
Center # : R6398-0A0

Cost share #: E-24-356
Center shr #: F6398-0A0

Rev #: 0
OCA file #:
Work type : RES
Document : GRANT
Contract entity: GTRC

Contract[redacted]
Prime #:

Mod #:

Subprojects ? : N
Main project #:

Project unit: ISYE Unit code: 02.010.124
Project director(s):
[redacted]

Sponsor/division names[redacted] / GENERAL
Sponsor/division codes:[redacted] / 000

Award period: [redacted]

Sponsor amount	New this change	Total to date
Contract value	95,000.00	95,00C 00
Funded	95,000.00	95,000.00
Cost sharing amount		4,750.00

Does subcontracting plan apply ?: N

[redacted]

PROJECT ADMINISTRATION DATA

OCA contact:[redacted] 894-4820

Sponsor technical contact	Sponsor issuing office
MICHAEL P. POLIS	JOE CARRABINO
(202)357-9618	(202)357-9602
NATIONAL SCIENCE FOUNDATION	NATIONAL SCIENCE FOUNDATION
ENG/ECS	DGC/ENG
WASHINGTON, DC 20550	WASHINGTON, DC 20550

Security class (U,C,S,TS)[redacted]
Defense priority rating : NONE
Equipment title vests with: Sponsor

ONR resident rep. is ACO (Y/N[redacted]
NSF supplemental sheet
GIT X

Administrative comments -
PROJECT INITIATION

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 05/28/90

'roject No. E-24-603 Center No. R6398-0A0

'roject Director WEISS G School/Lab ISYE

;ponsor NATL SCIENCE FOUNDATION/GENERAL

:ontract/Grant No. ECS-8712798 Contract Entity GTRC

'rime Contract No.

'itle TURNPIKE HEURISTICS FOR STOCHASTIC SCHEDULING & CONTROL OF QUEUEING NETWO

ffective Completion Date 900228 (Performance) 900531 (Reports)

loseout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	N	
Final Report of Inventions and/or Subcontracts	N	
Government Property Inventory & Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

CommentsNEGATIVE INVENTIONS INDICATED ON 98A.

ubproject Under Main Project No.

ontinues Project No.

istribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other	N
	N

NATIONAL SCIENCE FOUNDATION
Washington, D.C. 20550

FINAL PROJECT REPORT
NSF FORM 98A

PART I—PROJECT IDENTIFICATION INFORMATION

1. Institution and Address	2. NSF Program	3. NSF Award Number
Georgia Tech Research Institute Georgia Institute of Technology Atlanta, GA 30332-0420		ECS-8712798
	4. Award Period From 9/15/87 To 2/28/90	5. Cumulative Award Amount $95,000

6. Project Title
Turnpike Heuristics for Stochastic Scheduling and Control of Queueing Networks

PART II—SUMMARY OF COMPLETED PROJECT *(FOR PUBLIC USE)*

Consider scheduling a batch of jobs with stochastic (i.e. imprecisely known) processing times, on a set of parallel machines, with minimization of expected weighted flowtime (equivalently weighted average waiting time or weighted sum of completion times) as objective. We have shown that Smith's rule - start jobs with highest ratio of weight (waiting cost rate) over expected processing time first, is nearly optimal in two senses: The expected difference in the flowtime as well as the expected number of jobs for which Smith's rule is not optimal remain bounded, as the number of jobs.increases. We call these results approximate and turnpike optimality of Smith's rule.

Our results differ from most other results in stochastic scheduling in two major ways - they concern a heuristic rather than an optimal policy (note - the above problem is NP-hard for deterministic data), on the other hand they hold for general processing time distributions rather than for a special model (e.g. exponential distributions).

The study of parallel machines is one step in our attempt to understand scheduling of simultaneous operations. We have some partial results for the more complex situation of machines in series (tandem queues), where we investigate optimal order of machines.

We also approached simultaneous operation by studying restless bandit processes. In standard bandit processes only one process is changing at any time while the others are frozen; In restless bandits all the processes are changing simultaneously. Peter Whittle has suggested a heuristic for restless bandits, which is based on the Gittins index optimal solution to the standard bandit problem, and conjectured that it is asymptotically optimal. We have disproved Whittle's conjecture in general, but found sufficient conditions for it to hold.

The model of restless bandits seems closely related to problems of control of queueing networks, through fluid or diffusion approximations.

PART III—TECHNICAL INFORMATION *(FOR PROGRAM MANAGEMENT USES)*

1. ITEM *(Check appropriate blocks)*	NONE	ATTACHED	PREVIOUSLY FURNISHED	TO BE FURNISHED SEPARATELY TO PROGRAM: Check (✓)	Approx. Date
a. Abstracts of Theses	✓				
b. Publication Citations		✓			
c. Data on Scientific Collaborators		✓			
d. Information on Inventions		✓			
e. Technical Description of Project and Results					
f. Other *(specify)*					

2. Principal Investigator/Project Director Name *(Typed)*	3. Principal Investigator/Project Director Signature	4. Date
Gideon Weiss		5/23/90

Part III

(b) Publication Citations:

1. Pinedo, M. and Weiss, G., "The Largest Variance First Policy in Some Stochastic Scheduling Problems," *Oper. Res.* 35, pp. 884–891, 1987.

2. Weiss, G., "Branching Bandit Processes," *Probabl. Engineering Information Sci.* 2, pp. 269–278, 1988.

3. Coffman, E.G., Hofri, M. and Weiss, G., "Scheduling Stochastic Jobs with a Two Point Distribution on Two Parallel Machines," *Probabl. Engineering Information Sci.* 3, pp. 89–116, 1989.

4. Weiss, G., "Approximation Results in Parallel Machines Stochastic Scheduling," *Ann. Oper. Res. Special Volume on Production Planning and Scheduling*, M. Queyranne editor, to appear.

5. Weber, R.R. and Weiss, G., "On an Index Policy for Restless Bandits," *J. Appl. Probabl.*, to appear.

6. Huang, C.C. and Weiss, G., "On the Optimal Order of M Machines in Tandem," *Oper. Res. Letters*, to appear.

7. Weiss, G., "Turnpike Optimality of Smith's Rule in Parallel Machines Stochastic Scheduling," submitted to *Math. Oper. Res.*

8. Weiss, G., "Heuristics for Stochastic Scheduling and the Control of Queueing Networks," a report for NSF Grantee Meeting, in *Supplement to Proceedings of NSF Design and Manufacturing Systems Conference*, Arizona State University, Tempe, Arizona, January, 1990.

2 reprints of 1–3, 8 and a copy of 4–7 technical reports are attached.

(c) Data on Scientific Collaborators:

1. Dr. R.R. Weber, Lecturer, Cambridge University Engineering Department, Management Studies Group, Mill Lane, Cambridge CB1 1RX, U.K.

Dr. Weber served as consultant to the project, and has visited Georgia Tech in March 1988 and in March 1989 for collaborative work.

2. C.C. Huang, Graduate Student, School of ISyE, Georgia Tech.

Mr. Huang is pursuing thesis research for a Ph.D. within the project. He is expected to graduate in Winter 1991.

(e) Technical Description of Project and Results:

Attached is the body of a proposal for further support, which has been funded for the period March 1990–November 1993. This proposal contains a detailed description of the project and the results. For a more general survey of the research area see also the report for the NSF Grantee meeting, January 1990, attached as (a8).

PART IV - SUMMARY DATA ON PROJECT PERSONNEL

NSF Division ______________________

The data requested below will be used to develop a statistical profile on the personnel supported through NSF grants. The information on this part is solicited under the authority of the National Science Foundation Act of 1950, as amended. All information provided will be treated as confidential and will be safeguarded in accordance with the provisions of the Privacy Act of 1974. NSF requires that a single copy of this part be submitted with each Final Project Report (NSF Form 98A); however, submission of the requested information is not mandatory and is not a precondition of future awards. If you do not wish to submit this information, please check this box ☐

Please enter the numbers of individuals supported under this NSF grant.
Do not enter information for individuals working less than 40 hours in any calendar year.

*U.S. Citizens/ Permanent Visa	PI's/PD's		Post-doctorals		Graduate Students		Under-graduates		Precollege Teachers		Others	
	Male	Fem.	Male	Fem.	Male	Fem.	Male	Fem.	Male	Fem.	Male	Fem.
American Indian or Alaskan Native												
Asian or Pacific Islander					1							
Black, Not of Hispanic Origin												
Hispanic												
White, Not of Hispanic Origin	2				2		1					
Total U.S. Citizens					2							
Non U.S. Citizens	2				1							
Total U.S. & Non-U.S. ..	2				3		1					
Number of individuals who have a handicap that limits a major life activity.												

*Use the category that best describes person's ethnic/racial status. (If more than one category applies, use the one category that most closely reflects the person's recognition in the community.)

AMERICAN INDIAN OR ALASKAN NATIVE: A person having origins in any of the original peoples of North America, and who maintains cultural identification through tribal affiliation or community recognition.

ASIAN OR PACIFIC ISLANDER: A person having origins in any of the original peoples of the Far East, Southeast Asia, the Indian subcontinent, or the Pacific Islands. This area includes, for example, China, India, Japan, Korea, the Philippine Islands and Samoa.

BLACK, NOT OF HISPANIC ORIGIN: A person having origins in any of the black racial groups of Africa.

HISPANIC: A person of Mexican, Puerto Rican, Cuban, Central or South American or other Spanish culture or origin, regardless of race.

WHITE, NOT OF HISPANIC ORIGIN: A person having origins in any of the original peoples of Europe, North Africa or the Middle East.

THIS PART WILL BE PHYSICALLY SEPARATED FROM THE FINAL PROJECT REPORT AND USED AS A COMPUTER SOURCE DOCUMENT. DO NOT DUPLICATE IT ON THE REVERSE OF ANY OTHER PART OF THE FINAL REPORT.

COVER SHEET FOR PROPOSALS TO THE NATIONAL SCIENCE FOUNDATION

02.107.000.89.408

FOR CONSIDERATION BY NSF ORGANIZATIONAL UNIT (Indicate the most specific unit known, i.e. program, division, etc.)	PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE
Div. of Design and Manufacturing Systems ENGINEERING DIRECTORATE	

SUBMITTING INSTITUTION CODE (If known)	FOR RENEWAL ☐ CONTINUING AWARD ☐ ACCOMPLISHMENT BASED RENEWAL ☒ REQUEST, LIST PREVIOUS AWARD NO.:	IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? Yes X No __ IF YES, LIST ACRONYM(S)
0015693000	ECS-8712798	ARO - AFOSR - ONR

NAME OF SUBMITTING ORGANIZATION TO WHICH AWARD SHOULD BE MADE (INCLUDE BRANCH/CAMPUS/OTHER COMPONENTS)
GEORGIA TECH RESEARCH CORPORATION

ADDRESS OF ORGANIZATION (INCLUDE ZIP CODE) GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA 30332-0420

IS SUBMITTING ORGANIZATION: ☐ For-Profit Organization; ☐ Small Business; ☐ Minority Business; ☐ Woman-Owned Business

TITLE OF PROPOSED PROJECT
Heuristics for Stochastic Scheduling and the Control of Queueing Networks

REQUESTED AMOUNT	PROPOSED DURATION	DESIRED STARTING DATE
$260,320	Three Years	September 15, 1989

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW:

- ☐ Animal Welfare
- ☐ Endangered Species
- ☐ Human Subjects
- ☐ Marine Mammal Protection
- ☐ Pollution Control
- ☐ National Environmental Policy Act
- ☐ Research Involving Recombinant DNA Molecules
- ☐ Historical Sites
- ☐ Interdisciplinary
- ☐ International Cooperative Activity
- ☐ Research Opportunity Award
- ☐ Facilitation Award for Handicapped
- ☐ Proprietary and Privileged Information

PI/PD DEPARTMENT	PI/PD ORGANIZATION	PI/PD PHONE NO. & ELECTRONIC MAIL
Industrial & Systems Engineering	GEORGIA INSTITUTE OF TECHNOLOGY	(404) 894-3933

PI/PD NAME/TITLE	SOCIAL SECURITY NO.*	HIGHEST DEGREE & YEAR	SIGNATURE
Gideon Weiss, Professor		Ph.D. 1973	[signature]
ADDITIONAL PI/PD (TYPED)			
ADDITIONAL PI/PD (TYPED)			
ADDITIONAL PI/PD (TYPED)			
ADDITIONAL PI/PD (TYPED)			

For NSF Use:

TO BE COMPLETED BY THE AUTHORIZED ORGANIZATIONAL REPRESENTATIVE. By signing and submitting this proposal, the prospective grantee is providing the certifications set forth in (1) *Grants for Research and Education in Science and Engineering*, NSF 83-57 (rev. 11/87), and (2) Appendix C, 45CFR 620, Subpart F (Requirements for a Drug-Free Workplace).

(If answering *yes* to either, please provide explanation.)	YES	NO
Is the organization delinquent on any Federal Debt?		X
Is the organization presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency?		X

AUTHORIZED ORGANIZATIONAL REP. NAME/TITLE (TYPED)	SIGNATURE	DATE	TELEPHONE NO.
MATT GEDNEY, CONTRACTING OFFICER	[signature]	5/3/89	404/894-4817
OTHER ENDORSEMENT (optional) NAME/TITLE (TYPED)			

*Submission of social security numbers is voluntary and will not affect the organization's eligibility for an award. However, they are an integral part of the NSF information system and assist in processing the proposal. SSN solicited under NSF Act of 1950, as amended.

NSF Form 1207 (3/89)

EURISTICS FOR STOCHASTIC SCHEDULING AND THE CONTROL OF QUEUEING NETWORKS

Project Summary

The use of stochastic models for scheduling problems far from making the problems arder, tends to make them more tractable. This is illustrated by our results on the cheduling of jobs on parallel machines with the weighted flowtime as objective riterion: While the deterministic problem is NP-hard, we proved that a simple, easily pplied heuristic (Smith's Rule) is very close to optimal in the general stochastic etting.

By continuing to focus on heuristics we plan to tackle three further stochastic cheduling problems of practical importance: Scheduling of parallel machines subject o due dates, preemptive scheduling of parallel machines, and scheduling of machines in eries (flowshops). We shall again attempt to use the least restrictive assumptions on rocessing time distributions.

A deeper level of understanding for the dynamic optimal simultaneous operation of everal machines is suggested in the new "Restless Bandits" model of Peter Whittle. hittle suggests a Lagrangian relaxation and a priority rule which generalizes the ittins' index of bandit problems as upper and lower bounds for an optimal policy. ther important recent results on control of queueing networks are those obtained by .M. Harrison and L.M. Wein, using Brownian control problems as approximations.

We hope to prove Whittle's conjecture that his bounds are asymptotically optimal, nd to discover a strong connection between Whittle's approach and that of Harrison and ein. Based on these results we should be able to derive some new heuristics and to rove their asymptotic optimality.

TABLE OF CONTENTS

I. THE PROPOSED RESEARCH

1. INTRODUCTION

1.0. PRELIMINARIES

This proposal is for the continuation of our research on "Turnpike heuristics for stochastic scheduling and control of queueing networks" funded by the NSF grant ECS-8712798 for two years, from September 1987. The outline of that predecessor research was presented in our February 1987 proposal (appendix A). Although the predecessor grant period still has eight months to run, we were successful beyond our hopes in our research so far: Of the four research topics outlined in the proposal we completed and significantly extended the first topic, and we made substantial progress on topics 3 and 4, incorporating research on some new, unforeseen and exciting results of Peter Whittle which enhanced the significance of our program. We are satisfied with our general research direction and the current proposal reflects this continuity in direction, though our general outlook on the scope of the research has widened considerably.

The current proposal consists of Four parts. In the introduction we state our view of the research area, our general research philosophy and our broad interests and survey related material. Our progress on the research so far is reported next. This is followed by six research topics which we currently intend to pursue within the research area. We conclude the proposal by a section on expected impact.

We tried to make this proposal as self contained as possible, but, for brevity, we have avoided repetitions from the predecessor proposal. We attach the predecessor proposal for further background, as well as the referee reports, our progress report, and published papers and technical reports as appendices.

1.1 MOTIVATION AND GENERAL OVERVIEW

Scheduling problems of many types and forms occur in almost all fields of application. Typical examples are job shop scheduling in manufacturing, scheduling of hospital operating theaters in the service industries, production planning and project scheduling by management, ranking and selection of research and development projects, time sharing and software and hardware scheduling in data processing, and control of communication networks.

In recent years several stochastic scheduling models have been developed in addition to the older deterministic models, because they provide a more realistic and more general description of jobs, of policies and of problems. In stochastic scheduling, jobs are described by stochastic processing times. This implies unpredictable completion times and requires online decisions and dynamic scheduling policies, rather than static predetermined schedules. Our research addresses such problems.

The addition of random job arrival times extends stochastic scheduling to apply to queueing models. With arrivals, the single machine scheduling problem becomes the control of a single server queue, flowshops become tandem queues, and jobshops become queueing networks. We hope that our research will bridge some of the existing gaps between scheduling and the control of queueing systems.

Stochastic scheduling problems are in general very hard, and optimal solutions have so far been obtained only for very specific relatively simple models, usually under the assumption of some special probability distributions for the processing times, most notably exponential distributions. In order to handle practically meaningful problems we have from the outset decided to give up on the search for optimal solutions. Instead we wish to concentrate on heuristics, which are more tractable and which have the additional advantages that they require less data, require no distributional assumptions, and are more robust.

The study of heuristics has a twofold thrust - inventing heuristics, and assessing their performance. In looking for heuristics we focus on priority rules. Some of these are rather straightforward e.g. SEPT (sortest expected processing time first) or Smith's rule; others are much more sophisticated, e.g. Gittins index priority rules for bandit problems, and, most recently, index rules obtained by Lagrangean relaxation of "restless bandits". In assessing the performance of heuristics we are interested in two types of closeness to optimality:

- the expected value of the objective is close to optimal; we call this approximate (value) optimality.
- the actions taken are optimal most of the time; we call this turnpike optimality.

We have tried so far to tailor our research methods and theoretical tools to apply to as wide a range of applications as possible. The general type of applications we have in mind are stochastic flow systems through which several classes of jobs move, while being operated on singly by independent but simultaneous processors. Our aim is to obtain (perhaps somewhat weaker) results which apply to a wide range of these problems, in preference to stronger results that apply to only a few.

1.2. FORMULATION AND OUTLINE

Extensive progress has been made in stochastic scheduling over the last two years and a large number of papers extending existing results and looking at new problems have appeared. Our own results presage a new direction in that research, by presenting a worst case analysis of a heuristic for a stochastic scheduling problem. Significantly, we have addressed a problem whose deterministic version is NP-hard, and we have dispensed with all the assumptions usually made on distributions - our results hold for any processing time distributions. Also, as we predicted (predecessor proposal topic 1), the worst case performance of the heuristic (Smith's rule) in the general stochastic case is much better than in the corresponding deterministic problem. We believe that our general approach can be used to obtain heuristics with similar good worst case behaviour for several other stochastic scheduling problems of practical importance. These include parallel machines with arrivals, parallel machines with due dates, and preemptive scheduling of parallel machines (research topics 1 and 2).

One motivation for our research was to gain insight on systems which operate simultaneously on several stochastic jobs. Our results do indeed give a clear picture of how stochastic jobs are processed by parallel machines. We shall try to obtain similar insight into the other (much more complex) basic model of simultaneous operation, namely machines in series - flowshops or Tandem queues (research topic 3).

Our deeper and more ambitious research goal has been to bridge the gap between stochastic scheduling and the control of queues and queueing networks. Two important recent developments in the field may contribute towards this goal: Peter Whittle's restless bandits [Whtl88], and Harrison and Wein's heuristics for control of queueing networks based on the solution of Brownian control problems [HaWe87,88, Wein87,88b,c,d,89] In the predecessor proposal we have stressed bandit processes as an important paradigm for stochastic scheduling and the control of queueing networks (predecessor proposal research topic 3, current proposal research topic 4). Our "branching bandit processes" [Weis88a] capture most of the features of a queueing network, with the notable exception that they allow only one of the processors (servers) of the network to be active at any time. In his "restless bandits" paper Whittle suggests a heuristic for the simultaneous operation of several standard bandit processes. We are currently trying to prove that these priority index heuristics are, as Whittle conjectures, asymptotically optimal (research topic 5). Further we hope to obtain priority index rules for the simultaneous operation of several branching bandit processes. Such priority rules would constitute a heuristic for the control of a queueing network (this is essentially what we proposed in our predecessor proprsal topic 4, see current research topic 6). We conjecture that these heuristic rules will bear a close resemblance to the heuristic rules recently developed by Harrison and Wein. In establishing the connection between the two approaches we also hope to be able to investigate the question of asymptotic optimality of the heuristics of Harrison and Wein, which is open at the moment (research topic 6).

From the technical point of view, research on deterministic scheduling models generally lies within the theory of combinatorial optimization; stochastic scheduling typically requires the methods of stochastic optimization (stochastic dynamic programming, Markov decision processes, optimal control) and of queueing theory. Our research so far, on scheduling of parallel machines, has included a mixture of combinatorial optimization and stochastic optimization techniques. The results constitute to our knowledge the first worst case analysis of a stochastic scheduling problem. We hope to analyze machines in series via a Markov renewal process - this will be an extension of our approach to parallel machines. We shall require some additional probabilistic techniques in our research on restless bandits and on queueing networks. To prove Whittle's conjecture we intend to use fluid approximations to restless bandits and then apply large deviation theory. To establish a connection between Whittle's and Harrison and Wein's aproaches we shall require a less crude diffusion approximation to restless bandits.

1.3 SURVEY OF RELATED RESEARCH

We survey research related to our new proposed areas of research and recently published work. Additional references are given in the survey and bibliography of the predecessor proposal.

(I) Deterministic and stochastic scheduling - static models.

A recent survey by Lawler, Lenstra, Rinnooy Kan and Schmoys [LLRS 89) gives a fair picture of research trends in the theory of deterministic scheduling. While optimal algorithms and complexity results continue to appear, e.g.: [FeGr86, GaTW88, DuLe89, LaMa89, SiWa89], the emphasis has shifted towards pseudopolynomial algorithms, heuristics, approximations and bounding methods; see for example - [CaPi88, AdBZ88], for some search and bound techniques on job shop problems, [MoSi87, PoWa87] for decomposition pseudopolynomial approach, and [HaSh88, VLAL88] for some heuristics, the latter paper employs simulated annealing. In evaluating such heuristics the emphasis seems to have shifted from worst case analysis [KaKy86] to probabilistic analysis [BrDo86, FrRi86,87, CoLR88]; the latter has the disadvantage of artificial modelling assumptions but seems to give a more realistic (and optimistic) idea on the performance.

There has been a proliferation of new results in stochastic scheduling in the last couple of years; an extensive though incomplete list is [PiWe85, KuWa85, BoFo86, FoSu86, WiPi86, WeVW86, Kaem87, PiWe87, CFGW87, Righ88, PiRa 88, Fros88, XuMK89]. All of these results with the notable exception of [WiPi86] and [WeVW86] make very strong distributional assumptions, e.g. exponential, nonoverlapping, two point discrete. We mention that [WeVW86] contains the remarkable result that SEPT (Shortest Expected Processing Time First) minimizes expected flowtime on parallel machines under the assumption that processing times of different jobs can be stochastically ordered.

All these papers on stochastic scheduling contain exact optimality results. With the exception of [Nago88] in which a probabilistic analysis of a heuristic is performed, we are not aware of any performance analysis of a heuristic for stochastic scheduling problems, prior to our own research on worst case analysis of Smith's Rule [CoHW89, Weis88b,c].

In the predecessor research period we investigated $P||\Sigma w_jC_j$ - minimization of (expected) weighted flowtime on parallel machines. Some of the relevant references on this topic are [Smit56] - formulating Smith's rule which is optimal on one machine, [EEIs64] - analysing Smith's Rule as a heuristic for parallel machines, [Sanh76] - a pseudopolynomial algorithm, [LeRB77, GaJo79] - NP-completeness proof, [KaKy86, WHLL88] - worst case deterministic performance analysis of Smith Rule, [Kaem87] - some stochastic scheduling results.

An interesting analysis of the worst case performance of a heuristic for $P||\Sigma w_jT_j$ - minimizing total weighted tardiness on parallel machines (deterministic) is given by Arkin and Roundy [ArRo88]; we discuss this in research topic 1.

(II) Flowshops and Tandem Queues.

Minimization of makespan on the 2 machine infinite buffer flowshop, $F2||C_{max}$ is solved by Johnson's rule [John54], the blocked 2 machine flowshop $F2|nowait|C_{max}$ is solved as a special $O(n^2)$ travelling salesman

[LLRS85]. Every other flowshop problem seems to be NP-hard (see [GaJS76, Rock84a,b]). A number of papers characterize solvable special problems and derive bounds and heuristics - a good overview and a unified approach to most of these results is given by Monma and Rinnooy Kan [MoRi83].

Stochastic versions of flowshop scheduling problems are even harder. Earliest results concerned F_2|exponential jobs|C_{max}, where Johnson's rule applies [Bagg70, CuDu73, Weis82]. Further results were obtained in [Pine82a,b, FoSu86, WiPi86, BoFo86]. While dealing with special cases and making strong distributional assumptions, these papers point at some reasonable rules of thumb for flowshops with or without blocking. Two aspects - optimal order of jobs and optimal order of machines are considered. Additional insight into the order of machines in series is obtained in papers discussing tandem queues by Wolff and others [TeWo74, Wolf82, PiWo82, GrWo88] by Whitt [Whit85] and recently by Wein [Wein88a]; see also [LaNe80, Taka85, SuDi86, ScAl87, BiTi88, DaFr88, MaSl88].

An important property of flowshops and tandem queues is reversibility: Ordering machines in reverse orders does not affect the throughput under some general conditions; this is discussed by Yamazaki et. al. [YaSa75, YaSK 76, YaKS85], Dattatreya [Datt78] and Muth [Muth79].

When processing (or service) times are exponential, an even stronger property holds - for any arrival stream, with infinite buffers, the system output is independent of the order of the machines; this was shown by Weber [Webe79], an alternative proof was given by Lethonen [Leth86], and an extension to blocking systems was found by Chao, Pinedo and Sigman [ChPS88].

Muth [Muth73,77,84] describes some of the structural relations between processing times, blocking times and idle times for machines in series, and derives some Markov renewal underlying processes. This motivates some of our research on topic 3.

Kelly [Kell82b] discusses asymptotic behaviour of flowshops, as the number of machines becomes large.

(III) Bandit problems.

Gittins' formulation of families of alternative bandit processes and his priority policy solution by the Gittins index [GiJo74, Gitt79], has solved Bellman's [Belm56] multiarmed bandit problem; see also [NaGi77, GiNa77]. It was several years before it was realized that the Gittins index also solves the problem of the control of the M/G/1 queue with nonhomogeneous customers [VaWB85]. Thus it subsumes all the results of Klimov [Klim75,Klim78], Sevcik [Sevc74], Harrison [Harr75], Tcha and Pliska [TcPl77] and Meilijson and Weiss [MeWe77]. Prolific literature on the topic has appeared since, including two books: [BeFr85] and Gittins' own book [Gitt89]. Some interesting developments are [Kell81, Glaz87], while [ChKa86, Kall86, KaVe87] discuss calculation of the index. Recent work by Karatzas [Kara84] and Mandelbaum [Mand86,87], is impressively technical and extends the theory to continuous time.

Whittle has been responsible for much of the development of bandit problem theory, by his alternative optimality proof of the Gittins index [Whtl80], and by his extension to open processes [Whtl81]. This was extended by Lai and Ying [LaYi88] and by Weiss [Weis88a], who consider more general arrival patterns. Recently Whittle has come up with the restless bandits model [Whtl88]. We think this is a most significant development; it motivates our research topic 5 and parts of topic 6.

Sometimes an intriguing question is whether a given problem and solution can be cast as a bandit problem with a Gittins priority index solution. We mention [Kell82a, Smit78, NaWe82, KaDe84, BrYe87,89]; we shall investigate this in research topic 4.

(IV) Stochastic Flow Systems, Queueing Networks and Brownian Networks.

The literature surveyed in (I,II) and parts of (III) is of somewhat limited scope. Deterministic scheduling problems as in (I,II) are formulated as static problems - once the data is given a single optimal schedule is sought. The stochastic problems are more dynamic in nature - since information accrues with time, decisions have to be made dynamically; nevertheless the problems in (I,II) are ordered in space and time in such a way that a dynamic programming Markov decision problem formulation is usually possible. Stochastic flow systems such as queueing networks are much more complex than that: Things which happen simultaneously are not well ordered (as they were e.g. for machines in parallel or machines in series), and complex feedback phenomena abound.

Literautre that deals with manufacturing systems from a more global view point is too wide and varied for us to attempt a comprehensive review and we mention just a few approaches we are aware of: Work of Gershwin and others [KiGe 83, Gers86,87] work by Kumar and others [AkKu86, BiKu88, PeKu89, KuSe89], work by Solberg, Nof and Barash [Solb77, NoBS79, Solb83], work by Ho, Suri and others [HoCa83, TsBA86, Ho87, CaHo87, SuZa88], work by Buzacott, Yao and Shatikumar [BuSh80,85, ShSu87, YaBu86], by Stecke and others [StMo85, ShSt86] and by Seidman and Schweitzer [ShSS84].

Queueing networks [Kell79, Walr88, GePu87, Whtl86], provide the most versatile models for analysis and control of manufacturing systems. Increasingly important tools in their study are diffusion approximations, as described by Iglehart and Whitt [IgWh70, Whit74], Lemoine [Lemo78], and Reiman [Reim84], and fluid approximations as described by Mitra and by Alan Weiss [Mitr88, Weis86].

Harrison and coworkers [Harr85,88, HaWi86a,b, HaMa86] have used diffusion approximations to formulate Brownian control problems which are analogous, and which approximate, problems of optimal control of queueing networks. Wein and Harrison have in a series of papers solved these Brownian network control problems [HAWE87,88, WEIN87,88a,b,c,d,89] and obtained through them some very promising heuristic rules for the control of the original queueing networks. Our research topic 6 is largely motivated by the work of Wein and Harrison.

(V) Software, workstations and expert systems.

A significant recent trend has been the appearance of software tools in the form of specialized workstations and expert system shells which bring much of the modern theory of queuing, production planning, and flexible manufacturing techniques within the grasp of practitioners. We mention a few of the available systems that we are aware of without quoting exact references. In the area of queueing networks there are QNA-queueing netowrk analyzer, developed by Ward Whitt, PANACEA developed by Debasis Mitra, Q+ developed by Ben Melamed, all of them from Bell Labs, MANUPLAN developed by Rajan Suri, Larry Ho, and Diehl, and QUANTRON developed by Austin Lemoine at Ford Aerospace. In production planning and manufacturing we mention XCELL developed by Conway and Maxwell of Cornell, COSMOS a production planning tool developed by Muckstadt and Jackson of Cornell, GINO a general optimization tool developed by Liebman, Lasdon, and Schrage and GENSCHED an expert system shell developed by John Gilmore of Georgia Tech Research Institute.

2. PROGRESS SO FAR

2.0 SUMMARY OF PROGRESS

In this part of the proposal we summarize our progress so far on research supported by the predecessor NSF grant ECS-8712798. In this preliminary section we list papers and reports, we cross reference them with the research topics, we summarize personnel participation, and we refer to the predecessor proposal reviewers' comments. A topic by topic more technical discussion follows.

The following papers and reports contain the research undertaken so far; we attach those in appendix D.

- Papers which predated the support period:

(1) "The Largest Variance First policy in some stochastic scheduling problems" (with Michael Pinedo). Operations Research 35 pp 884-891 (1987). This paper underwent minor revisions in the grant period.

(2) "Branching bandit processes". Probability in the Engineering and informational sciences 2 pp 269-278 (1988). This paper underwent some minor revisions in the grant period.

(3) "Scheduling stochastic jobs with a two point distribution on two parallel machines" (with Ed G. Coffman, Jr. and Micha Hofri). Probability in the Engineering and informational sciences 3 pp 89-116 (1989). This paper underwent a major revision with some new proofs in the grant period.

- New reports written during the grant period:

(4) "A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime" (with C.C. Huang, C.L. Li, S. Liu, M. Pinedo, J. Song).

(5) "Approximation results in parallel machines stochastic scheduling". Submitted to Annals of Operations Research special volume on Production and Planning edited by M. Queyranne.

(6) "Turnpike optimality of Smith's rule in parallel machines stochastic scheduling". Submitted to Mathematics of Operations Research.

- Work in progress:

(7) "Asymptotic optimality of an index policy for restless bandits" (with Richard R. Weber) in preparation, not attached.

Classified by research topics we have:

<u>Predecessor Topic 1 (parallel machine batch scheduling)</u>: Paper 1, reports 4, 5, 6.

<u>Predecessor Topic 2 (Preemptive Scheduling)</u>: Paper 3.

<u>Predecessor Topic 3 (Branching Bandit Processes)</u>: Paper 2.

<u>Predecessor Topic 4 (Control of queueing networks)</u>: Preliminary work in report 7.

The personnel working on this research included, beside myself, Dr. Richard R. Weber of Cambridge University, England, and three graduate students. Dr. Weber has visited Georgia Tech for two weeks in March 1988 and will revisit in April 1989. Work with him is on Whittle's restless bandits conjecture. The student Chieng Chiu Huang has been working on the research for over a year; he was working on worst case analysis of Smith's Rule (deterministic) and is currently working on flowshop scheduling. The student Richard Griffin has started work recently on some bandit problems, and the student Michael Cole is in initial stages - he will work on scheduling problems of AGV's.

For the sake of continuity I am attaching reviewers' reports on the predecessor proposal in appendix B. I was fortunate to have an outstanding set of referees. Many of their perceptive comments have served to guide me in my research and to rethink my goals. In hindsight I am very happy (as I am sure the referees will be) that some of their misgivings about the feasibility of the proposed research were proved wrong, and that our results so far correspond and exceed what we proposed. I also hope that our results so far and what we propose to do is of greater practical value than some of the referees were led to believe by the predecessor proposal.

2.1 PREDECESSOR TOPIC 1: PARALLEL MACHINES STOCHASTIC SCHEDULING

In our research on this topic we have completed all the goals set in the proposal, confirmed all our conjectures and in fact obtained wider results: We proposed to investigate the performance of the SEPT (shortest expected processing time first) heuristic for the expected flowtime objective; we did in fact obtain results for Smith's Rule heuristic with the expected weighted flowtime as objective. Weighted flowtime as an objective is more useful in applications [CoMM67]; we also note that the deterministic parallel machine weighted flowtime problem is NP-hard [LeRB77,GaJo79].

We consider n jobs requiring processing times $X_1,\dots,X_n$ distributed (independently) as $F_1,\dots,F_n$. These are processed by $M+1$ parallel identical machines. Let $C_1,\dots,C_n$ denote the completion times of these jobs (which are random, they depend on the random processing times and on the scheduling strategy). With these completion times we associate the objective $\Sigma\, w_jC_j$ (weighted flowtime). Smith rule is to start the jobs in the order $1,\dots,n$ where $w_1/E(X_1) \geq w_2/E(X_2) \geq \cdots \geq w_n/E(X_n)$.

In comparing SR - Smith Rule, to OPT - the optimal strategy we have shown [Weis88b]:

(1) $$E\,(\Sigma\, w_jC_j \,|\, SR) - E(w_jC_j \,|\, OPT) \leq \frac{M^2}{2(M+1)} \left(\frac{w}{\mu}\right)_{max} \bar{D}^2 .$$

Let L be the (random) number of times that an optimal strategy chooses to start a job not according to SR. We have shown [Weis88c]:

(2) $$E(L \,|\, OPT) \leq \frac{M^2}{2} \left(\frac{w}{\mu}\right)_{max} \bar{D}^2 \,/\, \left(\Delta\frac{w}{\mu}\right)_{min} \mu_{min}\, \delta^{(M)} .$$

In these bounds, $\bar{D}^2$, $\delta^{(M)}$ are quantities obtained from $F_1,\dots,F_n$. $\bar{D}^2$ is the supremum of the expected squared remaining processing times (tails) from these distributions; $\delta^{(M)}$ is the infimum of the expected minimal positive remaining processing times for any subset of M jobs; the other constants depend on the means and the weights and are self explanatory. These bounds have the following consequences: Assume that the distributions $F_1,\dots,F_n$ are uniformly bounded in some sense, so that the above bounds do not grow with n; then (1) and (2) yield asymptotic optimality of SR as $n \longrightarrow \infty$; we call (1) approximate (value) optimality, (2) turnpike optimality.

Note that we need only to assume that the distributions are uniformly bounded; one way to assure that is for example to assume that all the hazard rates are bounded from above and from below by $\overline{\lambda} > \underline{\lambda} > 0$; another example is discrete NBU random processing times with uniformly bounded μ's and σ's. We do not assume that the actual job processing times are bounded, and indeed as $n \longrightarrow \infty$ we will almost surely have jobs with unboundedly long and unboundedly short processing times.

In contrast to this asymptotic behaviour we note [KaKy86, WHLL88] that for the deterministic problem the worst case of the ratio $\Sigma\, w_j C_j|SR \,/\, \Sigma\, w_j C_j|OPT$ is ~1.20. We believe that this contrast between the worst case performance of a heuristic in the deterministic and in the stochastic case is typical and will be found in most scheduling problems; this seems to us to be of great practical importance.

Also of practical importance is the fact that our analysis did not require any specific assumptions on the form of the processing time distributions.

The results (1) and (2) are quite innovative:

(a) To the best of our knowledge (1) constitutes the first worst case analysis of a heuristic for a stochastic scheduling problem.

(b) The turnpike optimality indicated by (2) says essentially that the SR-heuristic provides the optimal scheduling decision most of the time. We know of no similar results in the (deterministic or stochastic) scheduling literature.

The derivation of (1) and (2) is quite straightforward, but it provides a great deal of insight into the problem - we outline some of the ideas and insights here:

A Markov Renewal Representation of Parallel Machine Processing.

Assume jobs are started in the order $J(1),\dots,J(n)$. Let T_{j-1} be the instant at which the processing of job $J(j)$ starts. Let $D_{1j-1} \le \dots \le D_{Mj-1}$ be the remaining processing times on the other machines at this time, ordered by size. Then [Weis88b] $\{T_j;\underline{D}_j\}$ form a Markov renewal process [CiOz87]. In fact,

$$T_j - T_{j-1} = \Delta_j = \min\{X_j, D_{1j-1},\dots,D_{Mj-1}\}$$

and $D_{1j},\dots,D_{Mj}$ are obtained by ordering $\{X_j, D_{1j-1},\dots,D_{Mj-1}\}$, discarding the smallest , and subtracting Δ_j from all the others.

If job processing times are iid from a distribution F then $\underline{D}_j$ has stationary distribution which is that of an ordered M-sample from the equilibrium distribution of F.

Decomposition of Weighted Flowtime:

It is easy to see [Weis88b,c] that

$$\text{(3)} \quad \sum_{j=1}^{n} w_j C_j = \frac{1}{M+1} \sum_{j=1}^{n} \Big(\sum_{k=j}^{n} w_{J(k)} \Big) X_{J(j)} + \frac{1}{M+1} \sum_{j=1}^{n} w_{J(j)} \Big(M\, X_{J(j)} - \sum_{i=1}^{M} D_{ij-1} \Big)$$

Note that the first summand is the weighted flowtime for a single machine, with $M+1$ fold speed. Hence the second summand is the effect of splitting the processing between $M+1$ parallel machines.

Let $S_j^2 = \frac{1}{M} \sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)} \Big(\sum_{i=1}^{M} D_{ij} \Big)^2$, $j = 0,\dots,n$, that is, S_j is the sample variance of the components of $\underline{D}_j$. It is easy to see [Weis88b,c] that

$$(4) \qquad 2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_{J(j)} D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1)\, S_o^2 - M(M+1)\, S_n^2.$$

Note that on the right hand side the only term which depends on the order $J(1),\ldots,J(n)$ is S_n.

Expected Weighted Flowtime, when Weights Equal Mean Processing Times

Let μ_j, σ_j^2 be the mean and variance of F_j, $j=1,\ldots,n$. Let Π be any nonpreemptive work conserving scheduling strategy. Using (3) and (4) we show [Weis88b,c] that

$$(5) \; E\Big(\sum_{j=1}^{n} w_j C_j \Big| \Pi\Big) = \frac{1}{2(M+1)}\Big\{\Big(\sum_{j=1}^{n} \mu_j\Big)^2 + \sum_{j=1}^{n} \mu_j^2\Big\} + \frac{M}{2(M+1)} \sum_{j=1}^{n} \mu_j^2 \Big(1 - \frac{\sigma_j^2}{\mu_j^2}\Big)$$

$$- \frac{M}{2} S_o^2 + \frac{M}{2} E(S_n^2 | \Pi)$$

In this formula, the first term, which is $O(n^2)$, is the weighted flowtime for a single machine with $M+1$ fold speed. It depends on first moments of the distributions only. The second term, which is $O(n)$, is the effect of splitting the processing between $M+1$ machines. It depends on the first moments and on the coefficients of variation of the distributions only. If the coefficients of variation σ_j^2/μ_j^2 are small (e.g. for deterministic jobs) then splitting increases the expected weighted flowtime. If σ_j^2/μ_j^2 are close to 1 (e.g. for exponential jobs) splitting has no effect. For σ_j^2/μ_j^2 larger than 1, (e.g. hyper-exponential or DFR jobs) splitting decreases the expected weighted flowtime. This corresponds to the heuristic observation that parallel processing or processor sharing is advantageous when jobs are over-variable.

The last two summands in (5) are $O(1)$ under the assumption of uniform boundedness of $F_1,\ldots,F_n$. They correspond to the initial conditions and to the terminal conditions of the machines. Most remarkably, the expected weighted flowtime depends on the strategy Π only through the last term. We have therefore succeeded in isolating end effects in the schedule, and to show that apart from these end effects the expected weighted flowtime is of very simple structure, and is independent of the strategy.

The special role of the weights $w_j = \mu_j$ (or $w_j = X_j$) has also been noted elsewhere [EaEI64, KaKy86, ArRo88].

2.2 PREDECESSOR RESEARCH TOPIC 2: PREEMPTIVE SCHEDULING.

We have obtained results which parallel those of topic 1, for the following very simple preemptive scheduling model: A set of identically distributed jobs are to be scheduled on two parallel processors. Each job requires an amount of processing X, where $X = 1$ with probability p, and $X = k+1$ with probability 1-p $(k > 1)$. This is a model for jobs which consist for example of an inspection (duration 1) and, with probability 1-p, a repair (duration k). The decision here is whether or not to preempt jobs after the initial time period. This problem is in fact quite hard to solve optimally. The natural heuristic (which is optimal for one machine) is to always preempt in problems with $p > \frac{1}{k}$, never to preempt in problems with $p < \frac{1}{k}$. We have shown approximate and turnpike optimality of these policies [CoHW89].

2.3 PREDECESSOR RESEARCH TOPIC 3: BRANCHING BANDIT PROCESSES.

Most of the work on this topic has been performed prior to the funding period [Weis88a].

We have considered the following model for a queueing network. There are several types of customers which move through several nodes of the system; combining these we have classes $i = 1, \ldots, L$ each representing a customer of one type at one of the nodes. When a customer in class i receives service he occupies the server for a random duration V_i, at the end of which a reward C_i is generated and the customer disappears, to be replaced by $N_{i1},\dots,N_{iL}$ descendant customers of types $1,\dots,L$. The descendants represent the new node to which the customer may move, possible splitting of the customer into several jobs, change of the customer's class, and external arrivals. Linking arrivals to the class which is served is an important feature of this model. Rewards are discounted.

We analyse this model under the very strong and unrealistic limitation that there is only a single active server in the system and so at any moment in time only a single customer, of a particular class, is being served, while all the other customers are frozen.

Under this limitation, the problem of how to schedule the single active server's work has a simple solution: We show how to calculate a priority order of classes, and prove that an optimal policy is to serve at any moment one of the customers in the highest nonempty priority class.

2.4 PREDECESSOR RESEARCH TOPIC 4: CONTROL OF QUEUEING NETWORKS.

Our purpose here was to extend the model of predecessor topic 3 to simultaneously operating servers. Whittle's paper on "restless bandits" [Whtl87] indicated how this could possibly be done. Following the appearance of this paper we have concentrated on proving Whittle's conjecture in this paper, and on the application of similar ideas to "restless branching bandit processes". This work is in progress and is described in research topics 5 and 6.

3. THE RESEARCH TOPICS

3.0 OVERVIEW

We divided the proposed research into six topics. The first three are within the scheduling area, the last three are on stochastic control of queues and queueing networks. Within these we are confident about topics 1 and 2. Topic 3 is new to us, we have some preliminary results but no great expertise. Topic 4 is outstanding in that it may lead to problems outside of queueing and scheduling. the work in topic 5 is in progress; we are fairly confident that we should get at least some partial results. We hope these will lead to results on topic 6 which is more speculative. There is a wealth of problems here which we mention, probably more than we can work on in three years. In addition, stochastic scheduling and control of queues are areas of vigorous research, and we believe will become even more so in the near future. We therefore foresee that we will get a great deal of stimulation from other researchers. I am confident that we will find much that is of interest and of importance (theoretical and practical) to work on.

3.1 RESEARCH TOPIC 1: PARALLEL MACHINES STOCHASTIC SCHEDULING.

We shall be looking at three problems which extend our results so far.

(i) Performance of Smith's Rule in an M/G/C system.

It is known that Smith's rule or, as it is called in the queueing literature, the "$c\mu$-rule" is optimal for minimizing the expected weighted holding costs in an M/G/1 queue with several classes of customers (this is a special case of the control of the M/G/1 queue, [VaWB85]). The "$c\mu$" rule is in general not optimal for C parallel servers, it is not even known whether it is optimal for an M/M/C queue with several customer types [KeYe85].

Comparison of the single server queue with C-fold speed to the queue with C servers again shows (as in the batch case) that a possible region in which "cμ" is not optimal is at the end of each busy period. On the other hand it seems that the "cμ" rule may be optimal if there is a large number of customers waiting in the system.

We conjecture that end effects of busy periods account for the non-optimality of the "cμ" rule. We therefore hope to prove for M/G/C, C>1:

Conjecture 1, approximate value optimality:

$$E\ (\text{holding cost per busy period} \mid \text{"c}\mu\text{"}) - E\ (\text{holding cost per busy period} \mid \text{OPT}) \leq K.$$

Conjecture 2, Turnpike optimality:

If there are more than N_o customers in the system, then "cμ" is optimal.

We conjecture that under suitable conditions K and N_o are independent of the busy period length and hence "cμ" converges to optimal as the traffic intensity approaches 1.

(ii) Scheduling parallel machines with due dates.

Arkin and Roundy [ArRo88] have examined the problem of scheduling deterministic jobs on parallel machines with due dates, where the objective is to minimize weighted tardiness. they assume that the weight of a job is proportional to its processing time; this is plausible in some cases, when holding costs relate to economies of scale. They propose the following family of heuristics: Let d_i be the due date of job i, X_i its processing time, $s_i = d_i - X_i$ its latest untardy start date; schedule jobs in increasing order of $\gamma s_i + (1-\gamma)d_i$, $0 < \gamma < 1$.

We noted the special role of using the (expected) processing times as weights in section 2.1 (predecessor topic 1). Arkin and Roundy use the special properties of this case to obtain bounds on the worst case performance of their τ-family of heuristics for the deterministic problem. We shall attempt to perform a worst case analysis for the stochastic problem.

(iii) Improved turnpike results.

Our results [Weis88b,c] have been to show that in the optimal schedule of a batch of jobs with stochastic processing times, the expected number of times that a job is started not according to Smith's rule is bounded by a constant, independent of the size of the batch.

It would be interesting to show that this result holds in the almost sure sense, rather than in expectation, that is to say, this number of times is bounded almost surely by a constant.

Our intuition is that the following is true: There exists an N_o (independent of batch size) such that if there are more than N_o jobs left to schedule it is optimal to start the next job according to Smith rule. This conjecture cannot be derived merely from our results on expected turnpike optimality. It would however follow directly from the extension to almost sure turnpike optimality.

Existence of such an N_o would be very useful for the analysis that we propose on scheduling the M/G/C queue.

Admittedly, we have not been able so far to prove existence of such an N_o or obtain almost sure bounds even for the simpler problem of preemptive scheduling in Coffman, Hofri and Weiss [CoHW89]. We nevertheless believe it to hold at least for many special cases.

3.2 RESEARCH TOPIC 2: PREEMPTIVE SCHEDULING ON PARALLEL MACHINES.

Preemptive scheduling of jobs on a single server, to minimise expected weighted flowtime, is solved by the following Gittins index priority rule [Gitt82] : Let a job require processing time X, distributed like F, and assume the job has already received processing for a time x, and is not yet finished; call x the age of the job. The Gittins index of the job, at age x, is:

$$\nu(x) = \max_{y>0} \frac{\int_x^{x+y} dF(t)}{\int_x^{x+y} (1-F(t))dt}$$

If jobs j= 1,...,n have processing durations $X_1,\ldots,X_n$, distributed like $F_1,\ldots,F_n$, and if $x_1,\ldots,x_n$ are the jobs' ages, and $w_1,\ldots,w_n$ are their weights, calculate $w_1\nu_1(x_1),\ldots,w_n\nu_n(x_n)$, and then schedule the job with highest value of $w_i\nu_i(x_i)$ first.

This rule is in general not optimal for parallel servers [CoHW89]. Our aim is to show that it provides a good heuristic; in other words, if jobs are started on the parallel machines according to the Gittins priority rule the schedule is close to optimal.

In [CoHW89, see also section 2.2] we proved approximate (value) and turnpike optimality for the simplest nontrivial case: 2 machines, identical jobs with X_i= 1 or k+1 with probabilities p, 1-p; preemption of a job is allowed only after 1 time unit.

To illustrate how difficult it is to analyse optimal policies (as opposed to a given heuristic) we pose the following question: In the context of Coffman, Hofri and Weiss, is it true that the optimal policy would never preempt a job during its (deterministic) last k time units. We tried very hard to answer this question. We have absolutely no idea how to approach it and cannot conjecture the answer.

Our analysis of the Gittins index heuristic will proceed in well defined stages, by looking at a sequence of problems of successively greater generality.

- Analyze the Coffman, Hofri, Weiss model with more than two machines.

- Analyze i.i.d. jobs with $X = 1$ with probability p, $X = 1+V$ with probability $1-p$ where $V > 0$ is a random "tail" job.

- Analyze i.i.d. jobs with $X = 1, k+1, k+\ell+1$ $(k, \ell > 1)$, with probabilities $p, q, 1-p-q$.

We shall use a combination of the methods of [CoHW89,Weis88b,c] described in sections 2.1, 2.2 for these problems; nevertheless none of these three problems is straightforward, and each may require its own "trick".

If we can get satisfactory answers for these models, we hope to be able to approach the general case of jobs with artibrary distributions, and preemptions allowed at any time, or at any integer time unit.

3.3 RESEARCH TOPIC 3: MACHINES IN SERIES - FLOWSHOPS AND TANDEM QUEUES.

There are two aspects to machines in series: In the deterministic scheduling literature they are called flowshops, in the queueing literature they are called tandem queues. Extensive literature exists on both; see section 1.3. Scheduling of stochastic jobs on machines in series is in an inbetween area between the two. Problems here are very hard, especially when more than two machines are involved. We will look at cases of both finite and infinite waiting space.

(i) Machines in series with blocking:

We assume M machines in series, each with a capacity of one job (finite buffers can be replaced with 0-processing time machines), and an infinite number of jobs in front of the first machine. When processing times are exponential, the system can be represented by a Markov chain; the state lists each machine as working, idle or blocked [LaNe80, MaS188]. The difficulty here is the large state space. An alternative approach suggested by Muth [Muth84] avoids some of this difficulty, and is not limited to exponential processing times.

Following Muth we define: Let T_j be the time when job j starts on machine 1. Define $0 = D_{1j} = D_{2j-1} \leq D_{3j-2} \cdots \leq D_{Mj-M+1}$ such that job $j-k+1$ starts on machine k at time $T_j - D_{kj-k+1}$. Let $\underline{D}_j = (D_{1j}, D_{2j-1}, \ldots D_{Mj-M+1})$. Then $\{T_j, \underline{D}_j\}_{j=0}^{\infty}$ form a Markov renewal process. This approach is somewhat analogous to our Markov renewal formulation of parallel machines in [Weis88b.c], see section 2.1.

So far we used this formulation to obtain some preliminary results on the optimal order of machines, when jobs are identically distributed. Our assumptions on distributions are fairly general - we assume that processing times on the different machines are LR ordered.

We will also investigate questions of ergodicity, and methods for calculating steady state behaviour of this Markov renewal formulatin.

<u>(ii) Machines in series, no blocking:</u>

We shall look at the processing of a batch of n jobs, with stochastic processing times, on M machines in series. This problem lies between deterministic flowshop scheduling and tandem queues: Processing times are stochastic, but there are no random arrivals. We want to obtain some qualitative results: E.g. - if one machine is slower than all the others, with mean μ, then the makespan is asymptotically $n\mu + O(1)$; if machines are balanced the makespan is asymptotically $n\mu + O(\sqrt{n})$. Such results are not prominent in the literature - we hope they will enable us to come up with good practical scheduling heuristics.

3.4 RESEAECH TOPIC 4: BANDIT PROBLEMS.

Much research has been done recently on bandit problems, extending the theoretical basis and putting it on a rigorous footing; we surveyed some of this work in the introduction, section 1.3.

We would like to further explore two directions: (i) new applications and related problems; (ii) a classification of problems according to the underlying Markovian structure.

In the first direction, we note that the proliferation of recent research failed to bring up many new applications: Most of the applied results were known prior to the Gittins index formulation. On the other hand, several new results have recently been obtained which appear to be related to the Gittins index but they do not fit into the current framework of bandit problems. Here are a few:

Problem 1: Jobs with wait related processing times.

Jobs are to be scheduled on a single machine to minimize makespan or flowtime. Initial processing times of the jobs are $X_1, \ldots, X_n$, but these times increase linearly at rates $\alpha_1, \ldots, \alpha_n > 0$ as the jobs wait for processing, so that if job i is started at time t it will require processing time $X_i + \alpha_i t$. Arrivals may also occur. The makespan problem has been solved [BrYe87,89], and the solution looks like a Gittins priority rule. However it is not clear how to formulate this problem as a bandit problem. The flowtime problem appears very hard.

Problem 2: A batch broadcasting problem.

Transmitters $1, \ldots, k$ receive requests for broadcast as Poisson streams with rates $\lambda_1, \ldots, \lambda_k$. At each decision moment a transmitter is chosen and a single broadcast is sent to all its requests simultaneously, lasting a time $\exp(\mu)$. How is the transmitter chosen to minimize average wait (or average number of requests in the system)? The solution appears to be: Serve longest queue first. It is not clear whether this can be formulated as a bandit problem.

Problem 3: Scheduling of a single repairman.

A series system has n components which stay "up" for exponential times with rates $\mu_1,\ldots,\mu_n$ and then require exponential repair times with rates $\lambda_1,\ldots,\lambda_n$. A single server is available to do the repairs, (preemptively) and it is desired to schedule him so as to maximize the steady state availability of the system. The optimal policy is to repair always the best component (smallest μ) [Smit78, KaDe84, NaWe82], the proof is surprisingly hard.

In a similar problem let the system be general with penalty cost rates for failed components. If all $\mu_1 = \ldots = \mu_n$ a priority solution seems to exist even for generally distributed repair times; a bandit problem formulation is not known.

Problem 4: Bandit processes with constant death.

Consider a standard bandit problem, consisting of n arms described by a family of alternative bandit processes $X_1(t),\ldots,X_n(t)$, with discount rate α. It is well known that this is equivalent to an undiscounted problem, in which at each decision moment there is a probability of α that this is the last decision, all arms "die" after this decision and the "game" ends. A different version is to assume that following each decision, each of the arms (or each of the unactivated arms) has a probability α of disappearing and never reapppearing, independent of all other arms. Similar problems were posed in the general dynamic programming context [Ross83]. We do not know whether this problem has a "nice" solution. Indepently disappearing bandits are a special, simple, case of "restlessness" - more general restless bandits are the subject of research topic 5.

The motive to investigate these four problems and related problems is twofold: We may improve our understanding of bandit problems which are basic to our research. We may also be able to genaralize some of the results on these problems which are important in practice.

Our second research direction is less well posed. In dynamic programming there is a classification of problems into discounted, positive, negative, and average reward. A similar classification can no doubt be made for bandit problems; in fact various results in the literature can be classified as applying to one or the other of these models. However there has been no consistent attempt to pursue this classification in bandit problems. In particular, we want to identify the differences between recurrent and transient systems. This is motivated by our topic 6.

3.5 RESEARCH TOPIC 5: RESTLESS BANDITS.

Whittle [Whtl88] has recently considered the following generalization for bandit problems: There are n alternative bandit processes with states described by $X_1(t),\dots,X_n(t)$. At any moment in time exactly m of these are made active, the remaining $n-m$ are passive. Both active and passive arms change states, according to $P(X_j(t+1)|X_j(t),\text{active})$, $P(X_j(t+1)|X_j(t),\text{passive})$ and collect rewards $g(X_j(t),\text{active})$, $g(X_j(t),\text{passive})$ respectively. The problem is to choose which arms to activate at any time to maximize expected long term average return. This generalizes the standard bandit problem in that m arms (and not just one) are activated, and in that passive arms are not frozen, but may change state and generate rewards. Unlike the standard bandit problem which has a priority type optimal policy, the restless bandit problem does not have a nice, easily structured solution in general.

Whittle proposes to relax the condition that m arms exactly are active and to require instead only that the long run average of the number of active arms, $m(t)$, be equal to m. The relaxed problem has a simple solution: All states are partitioned into active and passive, to provide the optimal solution to the relaxed problem. The method of obtaining this solution is to attach a Lagrange multiplier to the constraint, and view its value as a subsidy for keeping an arm passive.

Whittle also shows how to obtain a priority policy for this problem: By varying the value of the Lagrange multiplier (the subsidy) every state will cross at least once from passive to active; the value of the subsidy at which the state crosses serves as its index; the index policy activates at any time the m arms whose states have the highest indices.

The relaxed and the index solutions provide upper and lower bounds for the optimal solution. Whittle conjectured that if $m/n = \theta$ remains constant while $m, n \longrightarrow \infty$, the long time average reward per unit time per arm, for all three solutions, converges to the same value.

So far we have obtained some partial results on this conjecture. We have shown that the relaxed optimal value (the upper bound) converges to optimal. We have also shown that the index policy value (the lower bound) converges to optimal if all the arms are moving between only two possible states.

On the full problem we made some progress. We assume that all the arms are identical, moving between k possible states, and that transitions happen in continuous time. We let $(n, z_1, \ldots, z_k)$ denote the system state where n is the number of arms and z_i the fraction of arms in state i. We can now define a fluid approximation [Mitr88] to this problem, under either the relaxed or the index policy. The fluid approximation satisfies $\dot{z}(t) = Q^{*}z(t)$ in the relaxed policy case, $\dot{z}(t) = Q(z(t))z(t)$ in the index policy case, where Q^{*}, $Q(z(t))$ are Markov transition rate matrices under these policies. We show that both these sets of equations have a fixed point π^* which is unique and is the same for both. We also know that $z(t) \longrightarrow \pi^*$ as $t \longrightarrow \infty$ in the relaxed case.

What we still need to show is that $z(t) \longrightarrow \pi^*$ as $t \longrightarrow \infty$, for any starting point $z(o)$ in the index policy case. If that is proved then by the theory of large deviations [Alan Weis86] the two policies converge to each other.

The question of whether $z(t) \longrightarrow \pi^*$ (global asymptotic stability of the equations $\dot{z}(t) = Q(z(t))z(t)$) may however turn out to be very hard.

3.6 RESEARCH TOPIC 6: CONTROL OF QUEUEING NETWORKS.

Whittle's ideas on restless bandits capture the essential step in how to move from a single server to a multiserver problem. Our aim is to extend his results and make them apply to control of queueing networks.

It should be noted first that this application and/or generalization is not straightforward.

Consider scheduling n jobs on m parallel machines. Here m jobs are made "active" while the other n-m remain passive (and in this case frozen). If we relax the constraint of m servers, and allow $m(t)$ servers, the whole problem disappears: Clearly, make $m(t) = n$ until all jobs are finished, $m(t) = 0$ thereafter. The difficulty is that this is a transient MDP problem, not a recurrent one as in Whittle's work. If arrivals are allowed (scheduling of the M/G/C queue), the problem is the same: Use $m(t)$ = number of customers in system; keep $E(m(t)) = m$ by having $m(t) = 0$ during idle periods. We hope however to get results similar to Whittle's when the traffic intensity is close to 1. In that case the analysis via fluid approximations will be much too crude but we may be able to analyze the system through diffusion approximations.

A possible extension of Whittle's restless bandits is to include several constraints (rather than one constraint, on the number of active arms). These may enable us to model more realistic and complex queueing networks.

The possible use of diffusion approximations and the heavy traffic conditions point in the direction of Harrison and Wein's research [HaWe87,88, Wein87,88b,c,d,89]. We also note that the heuristic policies which we expect to get through our approach may resemble those of Harrison and Wein, by specifying priority ordering of classes. Although the approaches are quite different, results from both approaches may complement each other. For instance, we hope to be able to supply asymptotic optimality proofs by using our approach.

II. EXPECTED IMPACT.

1. PI ASSESSMENT OF EXPECTED IMPACT

The research which we propose is of theoretical, basic research nature. In formulating it we were motivated not so much by any single applied or theoretical problem but by our general assessment of what the broad issues and prospects in stochastic scheduling are. In this assessment we considered both the requirements from the shop floor and the scope and limitations of available theory.

We are well aware of an alternative, practice oriented approach to scheduling research. There the analyst enters at the plant level, studies the environment and identifies its needs, proposes a solution strategy, and builds a model which is tailored to these needs, designs implementation, develops software and goes into the cycle of performance evaluation and modification. The usefulness of this approach is universally agreed upon. One should not however lose sight of some of the advantages which the theoretical approach offers, namely greater generality, freedom from the host of idiosyncratic encumbering details involved in every specific practical environment, and greater flexibility in applying more advanced theory.

We see a threefold impact for our type of proposed research: In advancing the theory of scheduling; in providing the practitioner with new insights, simple rules, and useful paradigms; and in generating algorithms which make better use of a broader range of system characteristics. We outline these three aspects now.

From the theoretical aspect, we feel that the proposed research as well as our results so far are both significant and innovative and we expect them to have a large impact on research in the area. We have introduced worst case analysis of stochastic scheduling heuristics and have successfully employed techniques of combinatorial optimization in conjunction with stochastic modelling. We believe we will find new extensions to existing theory of bandit problems, notably in developing Whittle's restless bandits - this will have a general impact on the theory of stochastic optimization. We also think that our work will have significant connections with other current work on approximations to discrete event stochastic systems such as fluid approximations and diffusion approximations.

Next we hope to have some impact on practitioners. To illustrate this impact consider our results on near optimality of Smith's rule. If a scheduler is faced by a problem of scheduling jobs on parallel machines, with weighted flowtime (i.e. job dependent linear holding costs) as an objective, our result is that Smith's rule (which he will no doubt be using anyway) is near optimal. If processing times are only known approximately and can be treated as stochastic, Smith's rule is in fact even closer to optimal than in the deterministic case. He can therefore dispense with costly attempts to find the (NP-hard) optimal schedule. If he would care to delve deeper into our results he will be able to see where and when (close to the end of the schedule) to modify Smith's rule to get even closer to optimal. More important, our results tell the scheduler that he needs to consider only the mean predicted processing time of each job for his near optimal schedule- features such as distribution variance (which has a large effect on the system performance) need not be considered (or measured) when devising the schedule. If in addition to weighted flowtime there should be (as is always the case in practice) other objective considerations (e.g. due dates, production smoothing) the robustness of Smith's rule makes it a good basis for modification into an acceptable good heuristic, and our explicit formulas for the (approximate) weighted flowtime may help in the evaluation of such modifications.

Another type of impact on the practitioner is provided by new paradigms. We feel that bandit processes, branching bandit processes and restless bandits are useful paradigms for the scheduler to keep in mind in his modelling efforts.

Finally, we propose to obtain some algorithms to calculate priority ordering of jobs. These will hopefully use data which is readily available, e.g. mean processing times, arrival rates, routing information, but which has not often so far been utilized to algorithmically produce schedules.

We believe that our research is extremely timely now. Over the past two years some very promising techniques for production scheduling, optimal control of queueing networks, and control of communications networks have emerged. These tackle very hard problems at a level of sophistication well beyond that of techniques previously available. We hope that our research will fit in with these exciting new developments.

At the same time the appearance of expert systems and of specialized workstations in scheduling, production planning and queueing is revolutionizing the workmode of the practitioner. These workstations create a tremendous demand for optimization algorithms, which they can readily absorb, and they promise to shorten the gap between theory and practice in the area.

2. LETTERS OF SUPPORT FROM INDUSTRY

Letters of support from industry for the proposed research have been solicited by Dr. Ira Pence, Director of the Materials Handling Research Center at Georgia Tech. All the companies approached have expressed interest in the research, and a desire to be kept informed about its progress with a view towards closer cooperation. Communication with these companies will be kept via their membership in the MHRC.

Letters of support are attached in the following pages from:

- IBM
- Litton
- Motorolla
- Xerox.

International Business Machines Corporation

Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598
914/945-3000

April 14, 1989

Dr. Ira W. Pence Jr., Director
Material Handling Research Center
Georgia Institute of Technology
765 Ferst Drive N. W.
Atlanta, GA 30332-0205

Dear Dr. Pence,

Thank you for sending me a copy of DR. Gideon Weiss' proposal entitled "Heuristics for Stochastic Scheduling and the Control of Queueing Networks". Yes, he is addressing an important, unsolved aspect of manufacturing. Yes, we are interested in what he is doing.

The manufacturing process by nature consists of a series of stochastic operations whether a job shop, flow line or some combination of both. One of the major challenges facing manufacturing is the planning and scheduling of production in such an environment.

One approach to solving this problem would be to reduce the variability of the operations involved. Unfortunately, technologies and products are changing so rapidly that operations never have a chance to "settle down". In some industries a stable process may well be an obsolete one.

Given that manufacturing processes will continue to be of a stochastic nature, it is imperative we learn to "live with them". Dr. Weiss' proposed research should help us understand how to do this. It is my understanding that up to this time there has not been significant research in this area. Although there are many papers with "stochastic" and "scheduling" in their titles, they assume away most of the normal manufacturing line complexities.

Please keep us advised of Dr. Weiss' progress. There are several people here in Manufacturing Research who wish to follow his work.

Regards,

Robert R. Leavitt
Senior Engineer, Manufacturing Research

Litton
Industrial Automation

Material Handling Div.
2300 Litton Lane
Hebron, Kentucky
41048

606 586-9800

April 11, 1989

Professor Donald Gross, Director
Division of Design and Manufacturing Systems
Engineering Directorate
National Science Foundation
1800 G Street, NW
Washington, DC 20550

Dear Professor Gross:

The work Dr. Gideon Weiss is doing at Georgia Tech and has proposed to continue ("Heuristics for Stochastic Scheduling and the Control of Queuing Networks") will be of direct interest to us. The process of extracting material from an AS/RS is inherently stochastic, yet our customers need to have reliable scheduling models.

To the extent Dr. Weiss is successful we will be able to use his results, both internally and on behalf of our customers. This type of work deserves your support.

Sincerely,

Stephen L. Parsley, P.E.
Director
Management Information Systems

SLP/dlt0030T

cc: John Mueller

April 12, 1989

Professor Donald Gross, Director
Division of Design and Manufacturing Systems
Engineering Directorate
National Science Foundation
1800 G Street, NW
Washington, D.C. 20550

Dear Professor Gross:

Electronics manufacturing is marked by uncertainty in the length of time required in many of its operations. Thus, the stochastic scheduling models which Dr. Gideon Weiss proposes to study as part of his proposal research "Heuristics for Stochastic Scheduling and the Control of Queueing Networks" are of interest to Motorola and to the Corporate Manufacturing Research Center.

Given our thrust into our flexible manufacturing configurations for our factories and emphasis on quality, the problems of dynamic scheduling are becoming increasingly important.

We will be following Dr. Weiss' work via our membership in the Manufacturing Research Center and the Material Handling Research Center at Georgia Tech and will consider incorporating his heuristics in our internal scheduling algorithm.

Sincerely,

William M. Beckenbaugh, Ph.D.
Director
Corporate Manufacturing Research Center

WMB:mb

[EEIS64] Fastman, W.L., Even, S., Isaacs, I.M. (1964) Bounds for the optimal scheduling of n jobs on m processors. Management Science 11, 268-279.

[FeGr86] Federguen, A., Groenevelt, H. (1986) Preemptive scheduling of uniform machines by ordinary network flow techniques. Management Science, 32, 341-349.

[FoSu86] Foley, R.D., Suresh, S. (1986) Scheduling n nonoverlapping jobs and two stochastic jobs in a flow shop. Naval Res. Logist. Quart. 33, 123-128.

[FrRi86] Frenk, J.B.G., Rinnooy Kan, A.H.G. (1986) The rate of convergence to optimality of the LPT rule. Discrete Appl. Math. 14, 187-197.

[FrRi87] Frenk, J.B.G., Rinnooy Kan, A.H.G., (1987) The asymptotic optimality of the LPT rule. Mathematics of Operations Research 12, 241-254.

[Fros88] Frostig, E. (1988) A stochastic scheduling problem with intree precedence constraints. Operations Research 36, 937-943.

[GaJo79] Garey, M.R., Johnson, D.S. (1979) Computers and Intractability: a Guide to the Theory of NP Completeness. Freeman, San Francisco.

[GaJS76] Garey, M.R., Johnson, D.S., Sethi, R. (1976) The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1, 117-129.

[GaTW88] Garey, M.R., Tarjan, R.E., Wilfong, G.T. (1988) One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research 13, 330-348.

[GePu87] Gelenbe, E., Pujolle, G. (1987) Introduction to Queueing Networks, John Wiley, New York.

[Gers86] Gershwin, S.B. (1986) Stochastic scheduling and set-ups in manufacturing systems. Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications (K.E. Stecke, ed.) 431-442, Elsevier, Amsterdam.

[Gers87] Gershwin, S.B. (1987) A hierarchical framework for discrete event scheduling in manufacturing systems IIASA Workshop on Discrete Systems: Models and Applications, August 3-7, Sopron, Hungary.

[Gitt79] Gittins, J.C. (1979) Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society Series B 14, 148-177.

[Gitt82] Gittins, J.C. (1982) Forward induction and dynamic allocation indices. Deterministic and Stochastic Scheduling M.A.H. Dempster et al (eds). Reidel, Dordrecht 125-156.

[Gitt88] Gittins, J.C. (1989) Bandit Processes and Dynamic Allocation Indices. Wiley, New York

[Gijo74] Gittins, J.C., Jones, D.M. (1974) A dynamic allocation index for the sequential design of experiments. In Progress in Statistics European Meeting of Statisticians 1972 (J. Gani, K. Sarkadi, I. Vince, eds.) 241-266, Amsterdam, North-Holland.

[GiNa77] Gittins, J.C., Nash, P. (1977) Scheduling, queues, and dynamic allocation indices. Proceedings EMS, Prague, 1974, Czechoslovak Academy of Sciences, 191-202, Prague.

[Glaz87] Glazebrook, K.D. (1987) Sensitivity analysis for stochastic scheduling problems. Mathematics of Operations Research 12, 205-223.

[GrWo88] Greenberg, B., Wolff, R.W. (1988) Optimal order of servers for tandem queues in light traffic. Management Science, 34, 500-508.

[HaSh88] Hall, L.A., Shmoys, D.B. (1988) Jackson's Rule for One-Machine Scheduling: Making a Good Heuristic Better, Department of Mathematics, Massachusetts Institute of Technology, Cambridge.

[Harr75] Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimality. Operations Research 23, 270-282.

[Harr85] Harrison, J.M. (1985) Brownian Motion and Stochastic Flow Systems. John Wiley, New York.

[Harr88] Harrison, J.M. (1988) Brownian Models of Queueing Networks with Heterogeneous Customer Populations. In Stochastic Differential Systems, Stochastic Control Theory and Applications IMA 10. (W. Fleming and P.L. Lions, eds) 147-186, Springer-Verlag, New York.

[HaMa86] Harrison, J.M., Mandelbaum, A. (1986) Brownian Networks: A Class of Linear Stochastic Systems. In preparation.

[HaWe87] Harrison, J.M., Wein, L.M. (1987) Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network. Submitted for publication.

[HaWe88] Harrison, J.M., Wein, L.M. (1988) Scheduling metworks of queues: heavy traffic analysis of a simple open network. Submitted for publication.

[HaWi86a] Harrison, J.M., Williams, R.J. (1986) Multidimensional reflected brownian motions having exponential stationary distributions. Ann. Prob., to appear.

[HaWi86b] Harrison, J.M., Williams, R.J. (1986) Brownian Models of Open Queueing Networks with Homogeneous Customer Populations, in preparation.

[Ho87] Ho, Y.C. (1987) Performance evaluation and perturbation analysis of discrete event dynamic systems. IEEE Transactions on Automatic Control AC32, 7, 563-572.

[HoCa83] Ho, Y.C., Cao, X.R. (1983) Perturbation analysis and optimization of queueing networks. Journal on Optimization Theory and Applications 40, 4, 559-582.

[IgWh70] Iglehart, D.L., Whitt, W. (1970) Multiple channel queues in heavy traffic, I and II. Adv. Appl. Prob. 2, 150-177 and 355-364.

[John54] Johnson, S.M. (1954) Optimal two and three state production schedules with setup times included. Naval Res. Logist. Quart. 1, 61-74.

[Kaem87] Kaempke, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines. Journal of Applied Probability 24, 430-447.

[Kall86] Kallenberg, L.C.M. (1986) A note on M.N. Katehakis and Y.R. Chen's computation of the Gittins Index. Mathematics of Operations Research 11, 184-186.

[Kara84] Karatzas, I. (1984) Gittins indices in the dynamic allocation problem for diffusion processes. Ann. Probab. 12, 173-192.

[KaDe84] Katehakis, M.N., Derman, C. (1984) Optimal repair allocation ina series system. Mathematics of Operations Research 9, 615-623.

[KaVe87] Katehakis, M.N., Veinott, A.F. (1987) The multiarmed bandit problem: decomposition and computation. Mathematics of Operations Research 12, 262-268.

[KaKy86] Kawaguchi, T., Kyan, S. (1986) Worst case bound of an LRF schedule for the mean weighted flow-time problem. SIAM J. Comput. 15, 1119-1129.

[KeYe85] Kella, O., Yechiali, U. (1985) Waiting times in the non preemptive priority M/M/c queue. Communications Statistics Stochastic Models 1, 257-262.

[Kell79] Kelly, F.P. (1979) Reversibility and Stochastic Networks, Wiley, New York.

[Kell81] Kelly, F.P. (1981) Multiarmed bandits with discount factor near one: the Bernoulli case. The Annals of Statistics 9, 987-1001.

[Kell82a] Kelly, F.P. (1982) A remark on search and sequencing problems. Mathematics of Operations Research 7, 154-157.

[Kell82b] Kelly, F.P. (1982) The throughput of a series of buffers. Adv. Appl. Prob. 14, 633-653.

[KiGe83] Kimemia, J., Gershwin, S.B. (1983) An algorithm for the computer control of production in flexible manufacturing systems. IIE Transactions 15, 353-363.

[Klim74] Klimov, G.P. (1974) Time sharing service systems I. Theory of Probability and Applications 19, 532-551.

[Klim78] Klimov, G.P. (1978) Time sharing service systems, II. Theory of Probability and Applications 23, 314-321.

[KuSe89] Kumar, P.R., Seidman, T.I. (1989) Dynamic instabilities and stabilization methods in distributed real time scheduling of manufacturing systems. Technical report, Department of Electrical and Computer Engineering, University of Illinois, Urbana.

[KuWa85] Kumar, P.R., Walrand, J. (1985) Individually optimal routing in parallel systems. J. Appl. Probab. 22, 989-995.

[LaNe80] Latouche, G., Neuts, M.F. (1980) Efficient algorithmic solution to exponnential tandem queues with blocking. SIAM J. Algebraic Discrete Methods 1, 93-106.

[LaYi88] Lai, T.L., Ying, Z. (1988) Open bandit processes and optimal scheduling of queueing networks. Adv. Appl. Prob.

[LaMa89] Lawler, E.L., Martel, C.U. (1989) Preemptive scheduling of two uniform machines to minimize the number of late jobs. Operations Research, to appear.

[LLRS85] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., (eds) (1985) The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimizatio, Wiley, Chichester.

[LLRS89] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (1989) Sequencing and scheduling: algorithms and complexity. Handbooks in Operations Research and Management Science Volume 4: Logistics and Production Planning.

[Lemo78] Lemoine, A.J. (1978) Networks of Queues - A Survey of Weak Convergence Results. Management Science 24, 1175-1193.

[LeRB77] Lenstra, J.K., Rinnooy Kan, A.H.G., Brucker, P. (1977) Complexity of machine scheduling problems. Ann. Discrete Math. 1, 343-362.

[Leth86] Lethonen, T. (1986) On the ordering of tandem queues with exponiential servers. J. Appl. Probab 23, 115-129.

[Mand86] Mandelbaum, A. (1986) Discrete multi-armed bandits and multi-parameter processes. Probab. Th. Rel. Fields 71, 129-147.

[Mand87] Mandelbaum, A. (1987) Continuous multi-armed bandits and multi-parameter processes. Ann. Prob. 14, 1527-1556.

[MaSl88] Matsuo, H., Slaten, P.M. (1988) A convex cmbination approximation for tandem queues with blocking. Technical report, U. of Texas, Austin, TX.

[MeWe77] Meilijson, I., Weiss, G. (1977) Multiple feedback at a single server station. Stochastic Processes and their Applications 5, 195-205.

[Mitr88] Mitra, D. (1988) Stochastic fluid models. Performance '87 Courtois, P.J., Latouche, G. eds, Elsevier, North-Holland, 39-51.

[MoRi83] Monma, C.L., Rinnooy Kan, A.H.G. (1983) A concise survey of efficiently solvable special cases of the permutation flowshop problem. R.A.I.R.O. Recherche operationnelle/Operations Research 17, 2, 105-119.

[MoSi87] Monma, C.L., Sidney, J.B. (1987) Optimal sequencing via modular decomposition: characterizations of sequencing functions. Mathematics of Operations Research 12, 22-31.

[Muth73] Muth, E.J. (1973) The production rate of a series of work stations with variable service times. International Journal of Production Research 11, 155.

[Muth77] Muth, E.J. (1977) Numerical methods applicable to a production line with stochastic servers. In Algorithmic Methods in Probability, North-Holland/TIMS Studies in the Management Sciences 7 (M.F. Neuts, ed) 143.

[Muth79] Muth, E.J. (1979) The Reversibility Property of Production Lines, Management Science 25, 152-158.

[Muth84] Muth, E.J. (1984) Stochastic processes and their network representations associated with a production line queueing model. European Journal of Operations Research 15, 63-83.

[Nago88] Nagorka, P. (1988) Probabilistic analysis of a heuristic for two machine stochastic flowshops with no intermediate storage. Submitted to Operations Research.

[NaGi77] Nash, P., Gittins, J.C. (1977) A Hamiltonian approach to optimal stochastic resource allocation. Advances in Applied Probability 9, 55-68.

[NaWe82] Nash, P., Weber, R.R. (1982) Dominant strategies in stochastic allocation and scheduling problems. Deterministic and Stochastic Scheduling M.A.H Dempster et al (eds.) Reidel, Dordrecht, 343-354.

[NoBS79] Nof, S.V., Barash, M.M., Solberg, J.J. (1979) Operational control of item flow in versatile manufacturing systems. International Journal of Production Research 15, 479-489.

[PeKu89] Perkins, J., Kumar, P.R. (1989) Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems. IEEE Transactions on Automatic Control AC34, 139-148.

[Pine82a] Pinedo, M. (1982) Minimizing the expected makespan in stochastic flow shop. Operations Research 30, 148-162.

[Pine82b] Pinedo, M. (1982) On the optimal order of station in tandem queues. Applied Probability-Computer Science: The Interface II (R.L. Disney and T.J. Ott, eds) 307-325, Birkhauser, Boston.

[PiRa88] Pinedo, M., Rammouz, E. (1988) A note on stochastic scheduling on a single machine subject to breakdown and repair. Probab. Engnrg. Inform. Science 2, 41-50.

[PiWe85] Pinedo, M., Weiss, G. (1985) Scheduling jobs with exponentially distributed processing times and intree precedence constraints on two parallel machines. Operations Research 33, 1381-1388.

[PiWe87] Pinedo, M.L., Weiss, G. (1987) The 'largest variance first' policy in some stochastic scheduling problems. Operations Research 35, 884-881 (attached in appendix).

[PiWo82] Pinedo, M., Wolff, R.W. (1982) A comparison of tandem queues with dependent and independent service times. Operations Research, 30, 464-479.

[PoWa87] Potts, C.N., van Wassenhove, L.N. (1987) Dynamic programming and decomposition approaches for the single machine total tardiness problem. European Journal of Operations Research 32, 405-414.

[Reim84] Reiman, M.I. (1984) Open queueing networks in heavy traffic. Mathematics of Operations Research 9, 441-458.

[Righ88] Righter, R. (1988) Job scheduling to minimize expected weighted flowtime on uniform processors. Systems & Control Letters 10, 211-216.

[Rock84a] Rock, H. (1984) The three-machine no-wait flow shop problem is NP-complete. Journal of the Association for Computing Machine 31, 336-345.

[Rock84b] Rock H. (1984) Some new results in flow shop scheduling. Z. Operations Research 28, 1-16.

[Ross83] Ross S.M. (1983) Introduction to Stochastic Dynamic Programming. Academic Press, New York.

[Sanh76] Sanhi, S. (1976) Algorithms for scheduling independent tasks. Journal of the Association for Computing Machine 23, 116-127.

[Sevc74] Sevcik, K.C. (1974) Scheduling for minimum total loss using service time distributions. Journal of the Association for Computing Machine 21, 66-75.

[ScAl87] Schweitzer, P.J. Altiok, T. (1987) Agregate modeling of Tandem queues with blocking. Technical report QM 8708, Rochester University.

[ShSS84] Shalev-Oren, S., Seidmann, A., Schweitzer, P.J. (1984) Analysis of flexible manufacturing systems with priority scheduling: PMVA. Proceedings 1st ORSA/TIMS Conference on Flexible Manufacturing Systems, University of Michigan, Ann Arbor.

[ShSt86] Shanthikumar, J.G., Stecke, K.E. (1986) Reducing work-in-process inventory in certain classes of flexible manufacturing systems, European Journal of Operations Research, to appear.

[ShSu84] Shanthikumar, J.G., Sumita, U. (1984) Approximations for the time spent in a dynamic job shop with application to due-date assignment. Working Paper No. 84-008, Department of Systems and Industrial Engineering, University of Arizona, Tucson.

[SiWa89] Simons, B., Warmuth, M. (1989) A fast algorithm for multi-processor scheduling of unit-length jobs. SIAM J. Comput., to appear.

[Smit78] Smith, D.R. (1978) Optimal repair of a series systems. Operations Research 26, 652-661.

[Smit56] Smith, W.E. (1956) Various optimizers for single stage production. Naval Res. Logist. Quart. 3, 59-66.

[Solb77] Solberg, J.J. (1977) A mathematical model of computerized manufacturing systems. Paper presented at the 4th International Conference on Production Research, Tokyo, August 22-30.

[Solb83] Solberg, J.J. (1983) Mathematical design tools for integrated production systems. In Efficiency of Manufacturing Systems (B. Wilson et al., eds) Plenum Press, New York.

[StMo85] Stecke, K.E., Morin, T.L. (1985) The optimality of balancing workloads in certain types of flexible manufacturing systems. European Journal of Operations Research 20, 68.

[SuDi86] Suri, R., Diehl, G.W. (1986) A variable buffer size model and its use in analyzing closed queueing networks with blocking. Management Science 32, 206-244.

[SuZa88] Suri, R., Zazanis, M.A. (1988) Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue. Management Science, January.

[Taka85] Takahashi, Y. (1985) A new type aggregation method for large Markov chains and its application to queueing networks. Teletraffic Issues in an Advanced Information Society M. Akiyama, ed., Elsevier, North-Holland

[TcPl77] Tcha, D.W., Pliska, S.R. (1977) Optimal control of single-server queueing networks and multi-class M/G/1 queues with feedback. Operations Research 25, 248-258.

[TeWo74] Tembe, S.V., Wolff, R.W. (1974) The optimal order of service in tandem queues. Operations Research 22, 824-833.

[TsBA86] Tsitsiklis, J.N., Bertsekas, D.P., Athans, M. (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Transactions on Automatic Control AC-31, 9, 803-812.

[VLAL88] VanLaarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K. (1988) Job shop scheduling by simulated annealing. Report OS-RSS09, Centre for Mathematics and Computer Science, Amsterdam.

[VaWB85] Varaiya, P., Walrand, J., Buyukkoc, C. (1985). Extensions of the multiarmed bandit problem: the discounted case. IEEE Transactions on Automatic Control AC 30, 426-436.

[Walr88] Walrand, J. (1988) An Introduction to Queueing Networks. Prentice Hall, New Jersey.

[Webe79] Weber, R.R. (1979) The interchangeability of /M/1 queues in series. Journal of Applied Probability 16, 690-695.

[WeVW86] Weber, R.R., Varaiya, P., Walrand, J. (1986) Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime. Journal of Applied Probability 23, 841-847.

[Wein87] Wein, L.M. (1987) Asymptotically optimal scheduling of a two-station multiclass queueing network. Unpublished Ph.D. dissertation, Department of Operations Research, Stanford University, Stanford, CA.

[Wein88a] Wein, L.M. (1988) Ordering tandem queues in heavy traffic. Technical report, Massachusetts Institute of Technology, Cambridge.

[Wein88b] Wein, L.M. (1988) Optimal Control of a two-station Brownian network. Mathematics of Operations Research, to appear.

[Wein88c] Wein, L.M. (1988) Scheduling networks of queues: Heavy traffic analysis of a two-station network with controllable inputs. Operations Research, to appear.

[Wein88d] Wein, L.M. (1988) Due date setting and priority sequencing in a multiclass M/G/1 queue. Technical report, Massachusetts Institute of Technology, Cambridge, Mass.

[Wein89] Wein, L.M. (1989) Scheduling network of queues: Heavy traffic analysis of a multistation network with controllable inputs. In preparation.

[Weis86] Weiss, A. (1986) A new technique for analysing large traffic systems. Advances Applied Probability 18, 506-532.

[Weis82] Weiss, G. (1982) Multiserver stochastic scheduling. Deterministic and Stochastic Scheduling (M.A.H. Dempster et al., eds.) 157-180, Reidel, Dordrecht.

[Weis88a] Weiss, G. (1988) Branching bandit processes. Probability Engineering Informational Sciences 2, 269-278 (attached in appendix).

[Weis88b] Weiss, G. (1988) Approximation results in parallel machines stochastic scheduling. Submitted to Annals of Operations Research special volume on Production and Planning (M. Queyranne, ed) (attached in appendix).

[Weis88c] Weiss, G. (1988) Turnpike optimality of Smith's rule in parallel machines stochastic scheduling. Submitted to Mathematics of Operations Research (attached in appendix).

[WHLL88] Weiss, G., Huang, C.C., Li, C.L., Liu, S., Pinedo, M. Song, J. (1988) A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime. Technical report (attached in appendix).

[Whit74] Whitt, W. (1974) Heavy Traffic Theorems for Queues: A Survey. In Mathematical Methods in Queueing Theory (A.B. Clarke, ed), Springer-Verlag, Berlin.

[Whit85] Whitt, W. (1985) The best order for queues in series. Management Science 31, 475-487.

[Whtl80] Whittle, P. (1980) Multiarmed bandits and the Gittins index. Journal of the Royal Statistical Society Series B42, 143-149.

[Whtl81] Whittle, P. (1981) Arm acquiring bandits. Annals of Probability 9, 284-292.

[Whtl86] Whittle, P. (1986) Systems in Stochastic Equilibrium. Wiley, New York.

[Whtl88] Whittle, P. (1988) Restless Bandits. J. Applied Probability to appear.

[WiPi86] Wie, S.H., Pinedo, M.L. (1986) On minimizing the expected makespan and flow time in stochastic flow shops with blocking. Mathematics of Operations Research 11, 336-342.

[Wolf82] Wolff, R.W. (1982) Tandem queues with dependent service in light traffic. Operations Research 30, 619-635.

[XuMK89] Xu, S., Mirchandani, Kumar, S.P. (1989) technical reports, not yet published.

[YaSa75] Yamazaki, G., Sakasegawa, H. (1975) Properties of duality in tandem queueing systems. Annals of the Institute of Statistical Mathematics 27, 201.

[YaSK76] Yamazaki, G., Sakasegawa, H., Kawashima, T. (1976) Production rate estimated with flow-shop reversibility (in Japanese). Trans. Japan Soc. Mechanical Engineers 43; Translated into English. Bulletin of the Japanese Society of Mechanical Engineers 21 (1978) 167.

[YaKS85] Yamazaki, G., Kawashima, T., Sakasegawa, H. (1985) Reversibility of tandem blocking queueing systems. Management Science 31, 1, Jan. 78-83.

[YaBu86] Yao, D.D., Buzacott, J.A. (1986) Models of flexible manufacturing systems with limited local buffers. International Journal of Production Research 24, 107.

Industrial and Systems Engineering

Report Series No. J-90-04

(Revision of J-88-09)

February, 1990

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

*This research was supported in part by NSF grant ECS-8712798.

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

Abstract

Consider scheduling a batch of jobs with stochastic processing times on parallel machines, with minimization of expected weighted flowtime as objective. Smith's Rule, which orders job starts by decreasing ratio of weight to expected processing time provides a natural heuristic for this problem. In a previous paper we have found an upper bound for the difference between the expected objective under Smith's Rule and under the optimal strategy. In this paper we find an upper bound on the expected number of times that Smith's rule differs from the optimal decision. Under some mild and reasonable assumptions these bounds remain constant when the number of jobs increases. Hence, under these conditions Smith's Rule is asymptotically optimal, and it has a Turnpike Optimality property, that is, Smith's Rule is optimal most of the time.

*This Research was supported in part by NSF grant ECS-8712798.

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

1. INTRODUCTION:

Many of the commonly occurring scheduling problems, in manufacturing, transportation, service and communications are notoriously difficult to solve optimally with deterministic data. (Lawler, Lenstra, Rinnooy Kan and Schmoys, 1989). Stochastic scheduling problems, in which data such as procesing times are subject to unpredictable, random fluctuations, are occasionally easier to solve than their deterministic counterpart (Weiss and Pinedo 1980, Pinedo 1983) - more often though their optimal solution is no less hard. Furthermore, the amount of data necessary to implement an optimal solution in the stochastic case is usually well beyond what is available in practice. It is for this reason that heuristics are of crucial importance in scheduling, both deterministic and stochastic.

A major challenge with heuristics is how to assess their effectiveness. Two approaches have been used to evaluate heuristics for deterministic scheduling problems: worst case analysis (Graham 1976, Kawaguchi and Kyan 1986), and average case (probabilistic) analysis (Frenk and Rinnooy Kan 1987). Worst case analysis gives an absolute bound on performance, and needs no assumptions; however, the worst case performance of a heuristic is often unacceptably bad, while the actual experience with the heuristic in practice may be excellent.

Probabilistic, average case performance, of a heuristic is usually a more realistic measure of how it will perform in practice; however, it requires an assumption on the probability distribution of the population of all problems, which clearly is not part of the model within which the heuristic performs: faced with an instant of the problem, the average case analysis has no relevance. These difficulties, are not a result of faulty or incomplete analysis; they seem to be inherent to the use of heuristics.

For some time I have thought that these difficulties are much less acute in stochastic scheduling. Informally speaking, there are several closely linked arguments that point in that direction: (a) in performing a worst case analysis of a heuristic for a stochastic scheduling problem, the expected performance is an average over the distribution of processing times for each instant. This averaging may yield a worst case performance which is similar to average performance in the deterministic case. (b) The anomalies and erratic behaviour of optimal schedules, as a function of small changes in the data, which are exhibited by deterministic scheduling problems (Graham 1976) are unlikely to occur with stochastic data. (c) Consider the gap between the optimal policy and a heuristic, for a deterministic problem. With less information the problem becomes stochastic and the performance of both the optimal and the heuristic policies deteriorates. However, the heuristic may be more robust than the optimal policy and the gap may close up. (d) There is a well known equivalence in some optimization problems between uncertainty and discounting. For discounted problems myopic heuristics usually perform very well. One could hope for ~~the~~ similarly good performance for stochastic problems.

The question is how to make use of these vague arguments to obtain concrete results. Our analysis shows how this can be done for the problem of scheduling stochastic jobs on parallel machines to minimize expected weighted flowtime, with Smith's rule as the heuristic.

We consider two aspects of the comparison between heuristic and optimal policy: the values of the objective functions, and the actions themselves. To compare the values of the objective functions we derive a bound on the difference between the expected value of the objective function under the heuristic and under the optimal policy. This bound is then used to obtain the worst case performance of the huristic. Under a wide range of plausible assumptions (see paragraph below), we obtain that the worst case performance ratio converges to 1, at a rate of $1 + O(1/n^2)$, as the number of jobs $n \longrightarrow \infty$; the determinstic worst case ratio in contrast is 1.20 (Kawaguchi and Kyan 1986, Weiss et al 1987). We refer to this result as "approximate optimality". To the best of my knowledge this result is one of the first worst case analyses of a heuristic for a stochastic scheduling problem (see Coffman, Hofri and Weiss 1989 for an earlier example).

To Compare the actions taken by the heuristic and by the optimal policy we derive a bound on the expected number of times that the heuristic may choose an action which is different from that taken by the optimal policy. Again, under a wide range of plausible assumptions this bound grows much more slowly than n, as the number of jobs $n \longrightarrow \infty$. We call this result "turnpike optimality". Loosely speaking, a heuristic has a turnpike optimality property if it takes the optimal action most of the time; the concept is used in dynamic programming (Shapiro 1968). To the best of my knowledge the use of the turnpike optimality concept in scheduling theory is quite novel (see Coffman, Hofri and Weiss 1989 for an earlier example).

Turnpike optimality may be of more importance than approximate optimality in scheduling theory. Often when a scheduling problem is solved, the schedule is implemented for only part of the time horizon, at which point unexpected changes necessitate resolving the problem; this cycle of implementing only the initial section of a solution and resolving with new data is repeated again and again throughout the plant operation time. Under these conditions turnpike optimality of the heuristic may imply that one is actually using the optimal action for the current data at any point in time.

Most of the exact optimality results in stochastic scheduling require very specific assumptions on the processsing time distributions: exponential distributions, finite support distributions, increasing hazard rate distributions, stochastically comparable distributions. An important feature of our results is that, since we talk only of approximate results, we do not need to make any assumptions about the processing time distributions. The bounds we derive on expected objective function difference, and on expected number of non optimal actions hold for any distributions. We do however need to assume some form of uniform boundedness on the distributions as the number of jobs $n \longrightarrow \infty$ to obtain asymptotic approximate optimality and turnpike optimality. Nevertheless, there is a huge difference between assuming uniform boundedness in the stochastic case and in the determinictic case: In the deterministic case, uniform boundedness means that all jobs are between some lower and upper bound, and one can not have arbitrarily long or short jobs as $n \longrightarrow \infty$. In contrast, in the stochastic case, uniform bounds on the distributions, for example in the form of uniform bounds on some moments, means no such thing - one can have arbitrarily short or long jobs when moments are bounded, and in fact as the number of jobs $n \longrightarrow \infty$, the longest and shortest jobs will as a rule approach zero and infinity respectively, which corresponds to real data behavior.

The proof of the approximate optimality result appears in an earlier paper (Weiss 1990). In the present paper we include the approximate optimality result (theorem 1) and the turnpike optimality result (theorem 2). We rederive the proof of theorem 1 (in a slightly strengthened form), both for the sake of completeness, and because the proofs of the two theorems are too intricately related to be able to present the turnpike theorem on its own. The earlier paper contains much additional material on the processing of stochastic jobs on parallel machines.

2. PROBLEM DESCRIPTION AND BACKGROUND.

A batch of n jobs (customers, tasks) is to be processed by M+1 identical parallel machines (servers, processors). Job j requires processing time X_j, to be provided by any one of the machines, where the value of X_j is specified by a probability distribution F_j, and the X_j's are drawn indepent of each other and of the schedule; a weight w_j is associated (as holding cost per unit time) with job j. Under some arbitrary scheduling rule let C_j be the completion time of job j; the cost of the schedule under this rule is the (random) weighted flowtime $\sum_{j=1}^{n} w_j C_j$.

The general problem of minimizing the expected weighted flowtime is intractable. A simple plausible heuristic is provided by Smith's Rule - order the job starts by decreasing weight to expected processing time ratio. For the unweighted flowtime (w_j=1, j=1,...,n), this rule boils down to SEPT - shortest expected processing time first.

A large body of research has been devoted to this problem and its various special cases. It is known that SEPT (Shortest Expected Processing Time First) and SR (Smith's Rule) minimize expected flowtime and weighted flowtime respectively, on a single machine (Smith 1956, Conway, Maxwell and Miller 1967). For parallel machines, SEPT minimizes expected flowtime in many important special cases, notably when for all i,j X_i and X_j are stochasticaly comparable (that is either $X_i \leq_{ST} X_j$ or $X_j \leq_{ST} X_i$, alternatively, either $F_i(x) \geq F_j(x)$ for all x or $F_j(x) \geq F_i(x)$ for all x - Weber, Varaiya and Walrand 1986).

Deterministic processing times, exponentially distributed processing times, and processing times which are distributed like tails of a single increasing hazard rate distribution are among the special cases of this remarkable result (McNaughton 1959, Bruno, Downey and Frederickson 1981, Glazebrook 1979, Weiss and Pinedo 1980, Weber 1982). In general however, SEPT may fail to be optimal (Pinedo and Weiss 1987, Coffman, Hofri and Weiss 1989). Minimization of weighted flowtime on parallel machines, for deterministic processing times X_j and general weights w_j, is an NP-hard combinatorial optimization problem (Lenstra, Rinnooy Kan and Brucker 1977, Garey and Johnson 1979), and very little can be said about optimal strategies for the minimization of expected weighted flowtime in the stochastic case (Kampke 1987).

The use of Smith's Rule as a heuristic can be quite inefficient - a worst case analysis shows that for deterministic processing times the ratio $R = (\Sigma\, w_j C_j \mid SR) / (\Sigma\, w_j C_j \mid OPT)$ can be as high as $1/2+\sqrt{1/2} \cong 1.20$ (Kawaguchi and Kyan 1986, Weiss et al 1987). This worst case occurs when the weights w_j are very nearly proportional to the processing times X_j, and it involves a very large number of very short jobs and $\cong 0.3\ M$ very long jobs.

Notwithstanding all these results, this paper was written to show that in a very general framework Smith's Rule provides a satisfactory solution to the problem.

The following intuitive discussion captures the essence of the problem: The argument in favour of SEPT (as well as for Smith's Rule), is that at the beginning of the schedule there is a large number of jobs waiting (the weighted cost rate of waiting jobs is high), and SEPT (or

Smith's Rule) tend to reduce the number of jobs (the cost rate of the waiting jobs) fastest. This argument suffices to prove optimality for a single processor, and it applies to parallel processors as well. For parallel processors there exists, however, a counter argument: Towards the end of the schedule, as jobs are completed, there are no more new jobs to start and the processors fall idle one after the other; this means that processing at the end becomes inefficient and this of course has an effect on the objective function.

To find the optimal solution requires (among other things possibly, though we believe that this is the only thing that needs to be considered) striking the right balance between reducing the cost rate fastest in a myopic way (Smith's rule), and between trying to plan in advance for a schedule completion which reduces the inefficiency periods to a minimum. Note that in some cases the right balance is to ignore the end effects, and use Smith's rule - for instance, as mentioned above, for unweighted flowtime and stochastically comparable jobs, SEPT is optimal. In contrast, if the weights are proportional to the expected processing times than Smith's rule is vacuous and the right balance is to concentrate only on the end effects. In this latter case, the minimization of the weighted flowtimes is equivalent to minimizing the sum of squares of the M+1 makespans (times at which the last jobs are finished) of the M+1 machines (see Weiss 1990). In particular, for 2 machines it is equivalent to minimization of the makespan of the last machine, C_{max}, which is NP-hard. Frenk and Rinnooy Kan (1987) show that LPT (longest processing time first) is a good heuristic for minimizing makespan in the deterministic case; similar arguments show it to be a good heuristic for reducing the inefficiency periods at the end of the schedule for our weighted flowtime problem. Striking the right balance is then, perhaps, choosing the time when to switch from Smith's rule to LEPT.

On the other hand the inefficiency at the end of the schedule is a boundary effect and is of marginal value. In particular under plausible assumptions its magnitude may remain bounded or at most grow only slowly as the number of jobs n, and the total length of the schedule, grow. Therefore, if, as we believe, this end inefficiency is the only reason that Smith's rule is not optimal, then we should be able to show that it is asymptotically optimal, and that asymptotically it is used by the optimal policy almost all the time. Indeed, in this paper, by studying the weighted flowtime on parallel machines we are able to quantify this end effect, find conditions under which it remains bounded as $n \longrightarrow \infty$, and thus get bounds on the expected difference in the objective function value between Smith's Rule and the optimal strategy, and bounds on the expected number of jobs which the optimal strategy does not schedule according to Smith's Rule.

3. THE MAIN RESULTS

For jobs $j = 1,\dots,n$ we assume the processing times X_j are nonnegative random variables drawn from distributions F_j. We let $\bar{F}_j = 1 - F_j$ and assume F_j have finite means μ_j and variances σ_j^2.

We require the following quantities, defined in terms of the processing time distributions:

$$\bar{D}^2 = \max_{1\le j\le n} \sup \left\{ \int_s^\infty (x-v)^2 \, dF_j(x) \,/\, \bar{F}_j(v) \;\middle|\; v \ge 0, \ \bar{F}_j(v) > 0 \right\} \qquad (1)$$

$$\delta^{(M)} = \min_{j_0,\dots,j_M} \inf\left\{ \int_0^\infty \bar{F}_{j_0}(x+v_0) \cdot\ldots\cdot \bar{F}_{j_M}(x+v_M) \, dx \,/\, \bar{F}_{j_0}(v_0) \cdot\ldots\cdot \bar{F}_{j_M}(v_M) \;\middle|\; v_0,\dots,v_M \ge 0, \ \bar{F}_{j_0}(v_0) > 0, \ \dots, \ \bar{F}_{j_M}(v_M) > 0 \right\} \qquad (2)$$

roughly these measure respectively how long and how short remainders of jobs can be: Consider looking at any of the jobs, at any time during its processing, when it is not yet completed, and take the expectation of the square of its remaining processing time; $\bar{D}^2$ is exactly the supremum of this quantity over all jobs and all times. Thus $\bar{D}^2$ provides for any busy machine at any time an upper bound on the ecpectation of the square of the time until the machine becomes available. Consider an instant at which k of the machines, $k \le M+1$, are busy, with a subset of k jobs that have received varying amounts ≥ 0 of processing up to that instant and have not yet been completed, and look at the expectation of the minimum of the remaining processing times of these jobs; $\delta^{(M)}$ is the infimum of this quantity over all subsets of jobs and all previous processing times. $\delta^{(M)}$ provides a lower bound on the expected time from the instant at which a macine becomes available until the next instant that a busy machine becomes available.

In addition we need the following quantities which depend also on the weights:

$$(\frac{w}{\mu})_{max} = \max_{1 \leq j \leq n} \frac{w_j}{\mu_j} \tag{3}$$

$$\mu_{min} = \min_{1 \leq j \leq n} \mu_j \tag{4}$$

$$(\Delta\frac{w}{\mu})_{min} = \min_{i \neq j} \left| \frac{w_i}{\mu_i} - \frac{w_j}{\mu_j} \right| \tag{5}$$

we assume that $\bar{D}^2 < \infty$ and $\delta^{(M)} > 0$.

We will consider two classes of strategies: Π_0 includes all nonpreemptive, work conserving (i.e. use no inserted idle time), nonrandomizing strategies which base decisions at time t on the history of the schedule up to time t. Π includes all nonpreemptive strategies which allow insertion of idle time as well as randomization and which base decisions at time t not only on the history of the schedule up to time t but also on the actual realized values of remaining processing times and inserted idle periods which occupy the machines at time t. In other words, when using strategies in Π, the processing times of jobs become known at the moment that their processing starts; similarly, the lengths of inserted idle periods are known at their start.

Assume now that $M+1$ machines are available to process jobs $1,\dots,n$ starting at time 0. The performance of Smith's rule ascompared to the optimal policy is summarized by the following two theorems:

Theorem 1 - Approximate Optimality (Weiss 1990): For any strategy $\pi \in \Pi$

$$E(\ \Sigma\, w_j C_j \mid SR\ ) - E(\ \Sigma\, w_j C_j \mid \pi\ ) \leq \frac{M^2}{2(M+1)} (\frac{w}{\mu})_{max}\, \bar{D}^2 \tag{6}$$

<u>Theorem 2 - Turnpike Optimality:</u> For any strategy $\pi \in \Pi_0$ such that

$$E(\Sigma w_j C_j \mid SR) - E(\Sigma w_j C_j \mid \pi) \geq 0 \qquad (7)$$

let L be the number of times that π starts a job not according to Smith's Rule. Then

$$E(L \mid \pi) \leq \frac{M^2}{2} (\frac{w}{\mu})_{max} \bar{D}^2 / (\Delta\frac{w}{\mu})_{min} \mu_{min} \delta^{(M)} . \qquad (8)$$

For practical purposes one is usually interested only in the smaller class of strategies, Π_0. However, we need Theorem 1 to hold for Π in order to prove Theorem 2 for Π_0. It is easy to construct counter examples to theorem 2 in the class of strategies Π. We take a close look at the two classes of strategies in Section 5.

The bounds provided by theorems 1,2 are ablolute, and hold for any set of weights and processing time distributions. The main interest in these bounds is when the number of jobs $n \longrightarrow \infty$. A favorable situation in that case is when the bounds in (6), (8) remain bounded as $n \longrightarrow \infty$. If the bound in (6) is bounded by a constant as $n \longrightarrow \infty$, and if μ_j, w_j are bounded from below, so that $E(\Sigma w_j C_j | \pi) = \Omega(n^2)$ then

$$R_1 = \frac{E(\Sigma w_j C_j |SR)}{E(\Sigma w_j C_j |OPT)} = 1 + O(\frac{1}{n^2})$$

Similarly, if the bound in (8) is bounded by a constant as $n \longrightarrow \infty$, then

$$R_2 = \frac{E(\#\text{ non SR opimal decisions})}{n} = O(\frac{1}{n}).$$

There exist many practical scenarios under which the values of $\bar{D}^2$ and of $\delta^{(M)}$, remain bounded for any number of jobs n. These include the following:

(a) If all the jobs belong to a finite number of types K, summarized by processing time distributions $F_1, \ldots, F_K$, and if $\bar{D}^2 < \infty$

and $\delta^{(M)} > 0$ for a set of jobs which includes $M+1$ jobs of each type, then $\bar{D}^2$ and $\delta^{(M)}$ are finite and remain fixed as n grows.

(b) If the hazard rate functions of the processing time distributions are all uniformly bounded by some upper bound value $\bar{\lambda}$ and some lower bound value $\underline{\lambda}$, then $\bar{D}^2 \le \frac{2}{\underline{\lambda}^2}$, and $\delta^{(M)} \ge \frac{1}{(M+1)\bar{\lambda}}$.

(c) If all the jobs have NBU (New Better than Used) type processing time distributions (X is NBU if $X-s|X>s$ is stochastically smaller than X for all $s>0$), then $\bar{D}^2 \le \max(\mu_j^2+\sigma_j^2)$, so it is enough to assume that μ_j and σ_j^2 are uniformly bounded for all n. The condition of NBU can be weakened to NBUE (X is NBUE if $E(X-s|X>s) \le E(X)$ for all $s>0$), and in addition, $E((X-s)^2|s>0) \le E(X^2)$ for all $s>0$.

(d) If the processing times have lattice probability distributions with a common time unit τ, that is all processing times are discrete random variables with values $\tau, 2\tau, \cdots$, then $\delta^{(M)} \ge \tau$.

In all these special cases it is still possible to have jobs which are long or short without any bound (with obvious exception of (d) for short jobs); the bounds are only on the probabilities.

4. APPROXIMATE OPTIMALITY OF SMITH'S RULE

In this section we summarize some of the results about expected weighted flowtime and prove Theorem 1. For more details and additional results see Weiss, 1990.

Consider the schedule obtained by starting the jobs in the order $I(1),\dots,I(n)$, without preemptions and without inserted idle time. In the formulation of theorems 1,2 we assume that all the machines become available at $t=0$. More generally, we now assume that the machines may become available at different times. Let $U_{00} \le \dots \le U_{M0}$ denote the ordered times at which the $M+1$ machines become available initialy - we shall assume throughout that $\sum_{i=0}^{M} U_{i0} = 0$. For $j = 1,\dots,n$ let $U_{0j} \le \dots \le U_{Mj}$ be the ordered times when the machines become available after the completion of jobs $I(1),\dots,I(j)$; the starting time of job $I(j)$ (the j'th job to start) is U_{0j-1}. We have the recursion:

$$(U_{0j},\dots,U_{Mj}) = \text{Order Statistics of } (U_{0j-1}+X_{I(j)},U_{1j-1},\dots,U_{Mj-1}) \quad (9)$$

For $j=0,\dots,n$, denote $D_{ij} = U_{ij} - U_{0j}$, $i = 1,\dots M$. Denote by S_j^2 the sample variance of $U_{0j},\dots,U_{Mj}$ (or of $0,D_{1j},\dots,D_{Mj}$)

$$S_j^2 = \frac{1}{M}\sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)}\left(\sum_{i=1}^{M} D_{ij}\right)^2. \quad (10)$$

It is easily seen that the completion of job $I(j)$ is at

$$C_{I(j)} = \frac{1}{M+1}\sum_{k=1}^{j} X_{I(k)} + \frac{M}{M+1} X_{I(j)} - \frac{1}{M+1}\sum_{i=1}^{M} D_{ij-1}. \quad (11)$$

Directly from (11) we get:

<u>Lemma 1:</u> The weighted flowtime can be decomposed as:

$$\sum_{j=1}^{n} w_j C_j = \frac{1}{M+1} \sum_{j=1}^{n} (\sum_{k=j}^{n} w_{I(k)}) X_{I(j)} + \frac{1}{M+1} \sum_{j=1}^{n} w_{I(j)} (M X_{I(j)} - \sum_{i=1}^{M} D_{ij-1}) \quad (12)$$

By examining the recursive relations between the D_{ij}'s the following key formula emerges:

Lemma 2: The job remainders D_{ij} are linked by the relation:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_{I(j)} D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1) S_0^2 - M(M+1) S_n^2 \quad (13)$$

Proof: For two machines we have $D_{1j} = |X_{I(j)} - D_{1j-1}|$; squaring and adding over j gives (13) for $M=1$. For $M>1$, (13) is obtained by applying the argument to each pair of machines, and summing over all pairs, see Weiss 1990 for more details. □

We now consider the special case of $w_j = \mu_j$. Taking expectations over (12) and (13), we have:

Lemma 3: For any nonpreemptive work conserving strategy $\pi \in \Pi$,

$$E(\sum_{j=1}^{n} \mu_j C_j | \pi) = \frac{1}{2(M+1)} (\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 \quad (14)$$
$$- \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2 | \pi)$$

Proof: The main step in the proof of Lemma 3 is to note that

$$E(X_{I(j)} D_{ij-1}) = E(D_{ij-1} E(X_{I(j)} | D_{ij-1}) = \quad (15)$$
$$= E(D_{ij-1} E(E(X_{I(j)} | I(j)) | D_{ij-1}) =$$
$$= E(D_{ij-1} \mu_{I(j)})$$

The second equality holds since D_{ij-1} is a function of $X_{I(1)}, \ldots, X_{I(j-1)}$ only, and so $E(X_{I(j)} | I(j), D_{ij-1}) = E(X_{I(j)} | I(j))$.□

The remarkable thing about formula (14) is that the only term in it which depends on the schedule π is $E(S_n^2)$; furthermore, the rest of the expression depends only on the first two moments of the distributions $F_1, \ldots, F_n$.

A direct corollary to Lemma 3 is:

Corollary 4: Let $\pi, \pi' \in \Pi$ be two strategies, and asssume that π is work conserving. Assume also that $S_0^2 = 0$. Then

$$E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi \right) - E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi' \right) \leq \frac{M}{2} E(S_n^2 \mid \pi) \leq \frac{M^2}{2(M+1)} \bar{D}^2 \quad (16)$$

Proof: By Lemma 5 (in the next section) it is possible to construct a strategy $\tilde{\pi}'$ which is randomizing but does not use inserted idle time, such that $I(1),\dots,I(n)$ have the same distribution under $\tilde{\pi}'$ as under π', and

$$E\left(\sum_{j=1}^{n} \mu_j C_j \mid \tilde{\pi}' \right) \leq E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi' \right)$$

Hence

$$E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi \right) - E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi' \right) \leq E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi \right) - E\left(\sum_{j=1}^{n} \mu_j C_j \mid \tilde{\pi}' \right)$$
$$= \frac{M}{2} E(S_n^2 \mid \pi) - \frac{M}{2} E(S_n^2 \mid \tilde{\pi}') \leq \frac{M}{2} E(S_n^2 \mid \pi) \quad (17)$$

By definition (1) and the assumption that $S_0^2 = 0$, we get $E(D_{ij}^2) \leq \bar{D}^2$ for all i,j which by definition (10) implies the second inequality in (16). □

From this point onwards we will assume that the jobs are ordered by Smith's Rule, that is $\frac{w_1}{\mu_1} \geq \dots \geq \frac{w_n}{\mu_n}$.

For any strategy we can rewrite the weighted flowtime as:

$$\sum_{j=1}^{n} w_j C_j = \frac{w_n}{\mu_n} \sum_{j=1}^{n} \mu_j C_j + \sum_{k=1}^{n-1} \left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right) \sum_{j=1}^{k} \mu_j C_j \quad (18)$$

It is now easy to obtain:

Proof of Theorem 1: We use (18) to write:

$$\Delta = E\left(\sum_{j=1}^{n} w_j C_j \mid SR\right) - E\left(\sum_{j=1}^{n} w_j C_j \mid \pi\right) \tag{19}$$

$$= \frac{w_n}{\mu_n}\left\{E\left(\sum_{j=1}^{n} \mu_j C_j \mid SR\right) - E\left(\sum_{j=1}^{n} \mu_j C_j \mid \pi\right)\right\} +$$

$$\sum_{k=1}^{n-1}\left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right)\left\{E\left(\sum_{j=1}^{k} \mu_j C_j \mid SR\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \pi\right)\right\}$$

We note that $\sum_{j=1}^{k} \mu_j C_j \mid SR$ and $\sum_{j=1}^{k} \mu_j C_j \mid \pi$ are the weighted flowtimes (with weights μ_j), of jobs $1,\ldots,k$, which are scheduled by SR in that order with no inserted idle time. Under π they are scheduled not necessarily in the order $1,\ldots,k$, and the schedule may include inserted idle periods, which consist of those jobs $j > k$ which π has started before the start of job k, as well as additional idle periods inserted by π prior to the start of job k. Nevertheless, the scheduling strategy provided by π for jobs $1,\ldots,k$ is in Π. Since SR is work conserving for every set $1,\ldots,k$, we can apply Corollary 4, and, noting that $\frac{w_n}{\mu_n} \geq 0$ and $\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}} \geq 0$, we get:

$$\Delta \leq \left\{\frac{w_n}{\mu_n} + \sum_{k=1}^{n-1}\left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right)\right\}\frac{M^2}{2(M+1)}\bar{D}^2 \tag{20}$$

which collapses to (6). □

5. DISCUSSION OF STRATEGIES

We take a closer look at strategies, starting with the wider and more complicated class of strategies Π. Consider a strategy $\pi \in \Pi$. To describe π we need to describe how a schedule is constructed by π. The construction of a schedule by π consists of a sequence of decisions on job or idle period starts, and updates of state. We use the following state desctiption: $s = (t,\underline{d},N,H)$ where t is the time; $\underline{d} = d_0,\dots,d_M$ is the machine availability vector, where $d_i = 0$ indicates that machine i is available at t while $d_i > 0$ indicates that machine i is not available at time t and will only become available at time $t + d_i$ – recall that strategies in Π can make use of the values of remaining processing times, in other words, the values d_i are deterministic and known; N is the set of as yet unstarted jobs, on which the current information is summarized by F_j the distribution of X_j for $j \in N$; Finally, H is the history of the schedule up to time t, which includes the start times of the jobs and the allocations of the machines over the period prior to t and also any additional information that π may use, excluding predictive information about the X_j, $j \in N$ – in particular H may include the information which is used by π to randomize decisions.

The initial state is $s = (t_0,\underline{d},N,H)$ where t_0 is the schedule start time, $\underline{d}$ are the initial machine availabilties, with $d_i = 0$ for at least one machine, $N = \{1,\dots,n\}$ includes all the jobs, and H is empty. At a decision time let the state be s, and at least one machine is available, with i_0 the available machine with lowest index. The decision will make machine i_0 unavailable for a time τ, to be determined by the decision. Based on the state s, and possibly using randomization, π will choose a job to start from N, or will choose an idle period. If an idle period is chosen, then τ will be the length of this idle period, which will be generated from some distribution G determined by π according to s. If job $j \in N$ is chosen to start then $\tau = X_j$ is generated from F_j. In either case, the value of τ becomes known immediately. Following the decision by π the state s is updated: let $\Delta t = \min(\tau, d_i, i \neq i_0)$, then $t := t + \Delta t$; $d_i := d_i - \Delta t$, $i \neq i_0$, $d_{i_0} := d_{i_0} + \tau - \Delta t$; $N := N - \{j\}$ if job j was chosen, $N := N$ if an idle period was chosen; finaly H is updated to indicate the allocation of machine i_0 and the job start if a job was started, and new information regarding future randomization may be added.

We shall add three provisos on decisions:

- If all the machines are available, a job has to start.

- We assume that for any state s, $\tau > 0$, and the time until the next decision, Δt, satisfies $E(\Delta t \mid \Delta t > 0) \geq \delta^{(M)}$ (in the case of randomization this has some complicated implications on G - however, we shall have no trouble verifying the condition for the schedules which we use in the proof).

- We assume that if $N = \phi$, an idle period of length $\tau = \max(d_0, \ldots, d_M)$ is inserted.

The schedule is complete if N is empty and all machines are available. The proviso that not all the machines can be idle assures that the length of the schedule is no more than $\sum_{j=1}^{n} X_j$ and therefore has finite expectation. The lower limit on $E(\Delta t|\Delta t>0)$, assures that the number of decisions taken to construct the schedule has a finite expectation. The third proviso makes the schedule complete at the completion of the last job.

The class of strategies Π_0 is much simpler than Π. The description is similar except that all the features regarding randomization and insertion of idle time are excluded.

The next Lemma on the construction of strategies with no idle times is crucial to the proof of Theorems 1 and 2.

Lemma 5: Let $\pi' \in \Pi$ be an arbitrary strategy which uses inserted idle time, then there exists a strategy $\pi \in \Pi$ such that π is randomizing but does not use inserted idle time, and such that the order in which jobs are started, $I(1),\ldots,I(n)$, has the same probability distribution under π' and π. Also, for all $j=1,\ldots,n$, $E(C_j|\pi) \leq E(C_j|\pi')$.

Proof: We prove the Lemma by describing the strategy π. We describe π by showing how a schedule is constructed by π. In addition to the state of the schedule constructed by π, which we denote by $s = (t,\underline{d},N,H)$, we shall make use of an auxilliary state, $s' = (t',\underline{d}',N',H')$, which will simulate the schedule constructed by π'. Initially we will have $t = t' = t_0$, and $s = s'$. Subsequently we shall use π' to update s' and use s' to update s. Starting from the initial states at time t_0 we will make a sequence of decisions and updates, in such a way that both s' and s get updated at every decision, and we have always $t' \geq t$, $N' = N$, and both $\underline{d}$ and $\underline{d}'$ have at least one available machine. Assume that following a sequence of decisions we have states s, s'. We use s' to obtain the decision taken by π', we generate a machine time τ, and we update s' accordingly. We then update s: If the decision of π' is to start an idle period, the state s remains unchanged. If the decision of π' is to start job $j \in N'$, we start job j on the machine with lowest index available in $\underline{d}$, we use the same value of X_j as realized in the update of s', that is we use a machine period $\tau = \tau'$, and we update s accordingly.

We note the following: Whenever s is updated, it is by assigning a job to an available machine; hence the schedule described by the state s uses no inserted idle time. At a decision state s, a sequence of one or more decisions which are taken by π' determines which job is chosen to start out of N; these decisions depend on the state s and the auxilliary state s' but are otherwise nonpredictive, thus the choice of job j is randomized but nonpredictive; the assumption of the lower bound $\delta^{(M)}$ assures that the randomization process has a finite expected number of steps. Once the job j is chosen, its processing time X_j is generated by F_j in the update of s. We let the state s describe the schedule generated by π; we have seen so far that π is a randomizing strategy using no inserted idle time, for scheduling the jobs $1,\ldots,n$. If we think of s' as part of the state history H, then it is clear that $\pi \in \Pi$. It is also clear from the construction that s' is a state which also describes a schedule, namely a schedule constructed by π'. Moreover, for every realization of this construction, the order in which jobs start, $I(1),\ldots,I(n)$, is the same for the two schedules described by s and by s'; this proves the statement that the job starting orders have the same distributions under both strategies. It remains to show that $E(C_j | \pi) \leq E(C_j | \pi')$. At any state s define $\underline{u}$ as the ordered vector of $t+d_0, t+d_1, \ldots, t+d_M$, and define $\underline{u}'$ analogously for s'. Initially, at $t=t'=t_0$, $\underline{u} = \underline{u}'$. We claim that thereafter, after each joint update of s' and s, $\underline{u} \leq \underline{u}'$. this is seen inductively: If π' chooses inserted idle time, only $\underline{u}'$ changes, and it has its smallest component increased and then all its components reordered; this can clearly not reduce any component; if π' chooses a job, the same τ is added to the smallest component of both $\underline{u}$ and $\underline{u}'$, and then they are both reordered, but then $\underline{u}'$ dominates $\underline{u}$ before reordering, and therefore also after the reordering. Since $\underline{u}'$ dominates $\underline{u}$ at every decision, job j starts earlier in s then in s', and, a-fortiori $E(C_j | \pi) \leq E(C_j | \pi')$. □

Immediately from Lemma 5, we have:

Corollary 6: The optimal schedule in Π does not use inserted idle time.

Corollary 7: Allowing randomization in Π or in Π_0 does not improve the optimal schedule.

Proof: Since the optimal policies use no idle time, the number of possible actions at a decision moment is at most n, and there are n decisions to be taken in the whole construction of the schedule. The proof is by backwards induction. □

6. TURNPIKE OPTIMALITY OF SMITH'S RULE:

The role of $\bar{D}^2$ and of $\delta^{(M)}$:

In the proof of Corollary 4 we saw that under every work conserving strategy π, if $S_0^{\;2} = 0$ then for all j, $E(S_j^{\;2} \mid \pi) \le \frac{M}{M+1}\bar{D}^2$; thus $\bar{D}^2$ puts an upper bound on the expected remaining processing times at each decision moment. The inequality (16) which is proved in corollary 4 was used in the proof of Theorem 1 and will also be used in the proof of Theorem 2.

On the other hand, consider the j'th decision moment of a strategy $\pi \in \Pi_0$ at which job ℓ starts its processing, and assume that all the other processors are occupied, so $D_{ij-1} > 0$, $i=1,\dots,M$. Then directly from the definition

$$E(\min(X_\ell, D_{1j-1}, \dots, D_{Mj-1})) \ge \delta^{(M)} \tag{21}$$

This puts a lower limit on the expected time between decisions, for every $\pi \in \Pi_0$. This is required in the proof of Theorem 2, but not in the proof of Theorem 1.

Proof of Theorem 2:

Let $\pi \in \Pi_0$ be a strategy which outperforms SR so that (7) holds. Rewriting (19) we have:

$$\begin{aligned} 0 &\ge E\Big(\sum_{j=1}^{n} w_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{n} w_j C_j \mid SR\Big) \\ &= \frac{w_n}{\mu_n}\Big\{E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{n} \mu_j C_j \mid SR\Big)\Big\} + \\ &\quad \sum_{k=1}^{n-1}\Big(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\Big)\Big\{E\Big(\sum_{j=1}^{k} \mu_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{k} \mu_j C_j \mid SR\Big)\Big\} \end{aligned} \tag{22}$$

Consider the jobs $1,\dots,k$ whose completion times appear in $E(\sum_{j=1}^{k} \mu_j C_j \mid \pi)$. The strategy π provides a srategy in Π for scheduling these jobs which possibly uses some inserted idle times in the form of processing times X_ℓ where job ℓ is started by π before job k, and $\ell > k$; π also uses randomization in scheduling jobs $1,\dots,k$, since their order may be determined by the values of the inserted jobs ℓ. For $k = 1,\dots,n-1$, let $\tilde{\pi}_{(k)}$ be the randomizing work conserving strategy for scheduling jobs $1,\dots,k$ which is constructed in Lemma 5 from π. We rewrite (22)

$$0 \geq \frac{w_n}{\mu_n}\{E(\sum_{j=1}^{n}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{n}\mu_j C_j \mid SR)\} + \tag{23}$$

$$\sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)}) - E(\sum_{j=1}^{k}\mu_j C_j \mid SR)\} +$$

$$\sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)})\}$$

We now apply corollary 4 to $\tilde{\pi}_{(k)}$ and obtain a bound on the first two summands as in the proof of Theorem 1. This gives us the following inequality which must hold for any π which outperforms SR:

$$\frac{M^2}{2(M+1)}(\frac{w}{\mu})_{max}\bar{D}^2 \geq \sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)})\} \tag{24}$$

We now obtain a lower bound on the difference between π and $\tilde{\pi}_{(k)}$.

Denote by $I_{\ell k}$ the indicator of the event that π starts job ℓ before job k - recall that SR is characterized by $I_{\ell k} = 0$ whenever $\ell > k$. Condition now on $I_{\ell k} = 1$ for some $\ell > k$, and on the state of the schedule constructed by π, at the instant that the service of job ℓ starts. Let $d_1,\dots,d_M$ be the availablility vector of all the other machines, it consists of remaining processing times of jobs which have had some known amount of service, and of the full processing time of jobs which start simultaneously with job ℓ. Let $\tau = \min(X_\ell, d_1,\dots,d_M)$.

Since $\pi \in \Pi_0$ is not predictive, the remaining processing times are distributed like $F_j(x+v)/\bar{F}_j(v)$ where v is the amount of processing already received by job j, and so the time until the earliest next job start has expected value of at least $\delta^{(M)}$. Removing the processing of job ℓ from the schedule will not increase any job completion time within jobs $1,\dots,k$, and at least one job (either k or some job k' with $k' < k$) will start earlier by τ. So the expected saving in the weighted flowtime (with weights μ_j) conditional on $I_{\ell k} = 1$, where $\ell > k$, is at least $\mu_{min}\ \delta^{(M)}$.

The saving by removing a job ℓ which π has inserted in front of job k appears to count only once for every k. However, it is easy to construct $\tilde{\pi}_{(k)}$ in several stages, starting from the schedule under π, and at each successive stage removing the last inserted idle job. Each such a removal will reduce the expected weighted flowtime of the sum over $1,\dots,k$ at least by the above amount. Note however, that if several idle jobs were inserted simultaneously, the removal of all of them from the schedule may have the same effect as to remove only one. So we can only assume an improvement of $\mu_{min}\ \delta^{(M)} / (M+1)$ in the expected sum over $1,\dots,k$ for each event $I_{\ell k} = 1$, where $\ell > k$.

We have thus shown:

$$E\left(\sum_{j=1}^{k} \mu_j C_j \mid \pi\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \tilde{\pi}_{(k)}\right) \geq \sum_{\ell > k} P(I_{\ell k} = 1)\ \mu_{min}\ \delta^{(M)}/(M+1) \tag{25}$$

Using the lower bound on $\left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right)$, and summing over $k=1,\dots,n-1$, we have:

$$\sum_{k=1}^{n-1} \left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right) \left\{E\left(\sum_{j=1}^{k} \mu_j C_j \mid \pi\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \tilde{\pi}_{(k)}\right)\right\} \geq \tag{26}$$

$$\sum_{k=1}^{n-1} \sum_{\ell > k} P(I_{\ell k} = 1)\ \left(\Delta\frac{w}{\mu}\right)_{min}\ \mu_{min}\delta^{(M)}/\ (M+1)$$

But

$$E(L \mid \pi) \leq \sum_{k=1}^{n-1} \sum_{\ell>k} P(I_{\ell k} = 1) \tag{27}$$

and substituing (26), (27) in (24) the theorem follows. □

6. DISCUSSION

We discuss briefly some related problems and directions for future research.

An explicit formula for expected weighted flowtime. Our results are based on various formulas which we derived for expected weighted flowtime. The most general form is obtained by substituting (14) into (18). We have, if jobs are started in the order 1,2,...,n:

$$E\left(\sum_{j=1}^{n} w_j C_j \mid M+1 \text{ machines}\right) = \frac{1}{M+1} E\left(\sum_{j=1}^{n} w_j C_j \mid \text{One machine}\right) + \frac{M}{2(M+1)} \sum_{j=1}^{n} w_j \mu_j \left(1 - \frac{\mu_j^2}{\sigma_j^2}\right) + \frac{M}{2} \sum_{j=1}^{n} \frac{w_j}{\mu_j} \{E(S_j^2) - E(S_{j-1}^2)\} \tag{28}$$

The first term is the weighted flowtime on a single, M+1 fold speed machine; it depends only on the first moments of the processing time distributions, it is of order $O(n^2)$, and it is minimized by SR. The second term is composed of per job delays caused by the parallel processing; it is a function of the means and the variances of the processing times, it is of order $O(n)$, and it is independent of the schedule. The last term is the intractable part which may depends on the full description of the processing time distributions as well as on the schedule. For large n it may however exhibit some limiting steady state behaviour.

Preemptive scheduling of a batch of jobs on parallel machines. Preemptive scheduling of a batch af jobs on a single machine, to minimize weighted flowtime, is optimized by using a Gittins index policy (Gittins 1979, 1982). On parallel machines this suggests to schedule at any moment the jobs with the highest Gittins index as a suboptimal heuristic, analogous to the use of Smith's Rule in the nonpreemptive case. A very special case of i.i.d jobs with a two point distribution on two parallel machines has been analyzed by Coffman, Hofri and Weiss 1989. It would be interesting to generalize these results.

Scheduling of a stream of arriving jobs. If jobs with various processing time distributions arrive at a single server in independent Poisson streams, then Smith's Rule and the Gittins index policy remain optimal (see Sevcik 1974, Klimov 1974, Harrison 1975, Meilijson and Weiss 1977, Gittins and Nash 1977). Using these rules for parallel servers provides suboptimal heuristics. Clearly, there is now a nonoptimal end effect at the end of each busy period, as well as inefficiencies within each busy period which arise whenever the number of jobs falls below the number of mechines. Nevertheless it may be possible to bound the worst case behavior of these heuristics.

Extensions to control of queueing networks. In Weiss (1988) Gittins type priority rules for scheduling customers in a queueing network which is served by a single server (the server jumps between the nodes of the network and provides preemptive service) are derived. These may provide some heuristics for more conventional networks in which all the nodes are served simultaneously by several servers in parallel.

Restless Bandits. Whittle (1988) has recently considered some generalizations of Gittins' original Bandit process model. In scheduling terms these can be expressed as including several parallel servers as well as exogenous changes in waiting jobs. Whittle suggests a Gittins' type heuristic for these processes, and conjectures that under the appropriate asymptotic conditions these may converge to optimal. Weber and Weiss (1990) discuss conditions under which the conjecture holds, as well as some counter examples. Our results in this paper provide a special case for which Whittle's conjecture holds.

Queueing network heuristics based on diffusion approximations. Recently Wein (1987) has derived some heuristics for the control of queueing networks by considering heavy traffic conditions and using diffusion approximations. Some parts of these heuristics appear to be priority type rules to schedule several types of customers. It is intriguing to try and find a possible connection between our current work, Whittle's conjecture, and Wein's results.

References:

1.Bruno, J., P. Downey and G. Frederickson, (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan, J. Assoc. Comput. Mach, 28, 100-113.

2.Coffman, E. G., Jr., M. Hofri, G. Weiss, (1989). Scheduling stochastic jobs with a two point distribution on two parallel machines. Probab. Eng. Inform. Sci., 3, 89-116.

3.Conway R.W., Maxwell W.L., Miller L.W. (1967) Theory of scheduling Addison Wesley,Reading MA.

_4.Frenk, J.B.G. and A.H.G. Rinnooy Kan (1987). The asymptotic optimality of the LPT rule. Math. Operat. Res. 12, 241-254.

5.Garey, M. R., D. S. Johnson (1979) Computers and Intractability: a Guide to the Theory of NP Completeness. Freeman, San Francisco.

6.Gittins, J. (1979) Bandit processes and dynamic allocation indices. J. Roy. Statist Soc. B. 14, 148-167.

7.Gittins, J. (1982) Forwards induction and dynamic allocation indices. In Deterministic And Stochastic Scheduling. (M.A.H Dempster et al, eds.) Reidel, Dordrecht, 125-156.

8.Gittins, J., P. Nash (1977) Scheduling, queues, and dynamic allocation indices. Proc. EMS, Prague, 1974. Czechoslovak Academy of Sciences, Prague, 191-202.

9.Glazebook, F. D. (1979) Scheduling tasks with exponential service times on parallel processors, J. Appl. Probab., 16, 685-689.

10.Graham, R.L. (1976) Bounds on the performance of scheduling algorithms, Computer and Job Shop Scheduling Theory, EG. Coffman Jr. editor, J. Wiley, New York, 165-227.

11.Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimaly. Oper. Res. 23, 270-282.

12.Kampke, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines, J. Appl. Probab., 24, 430-447.

13.Kawaguchi, T., Kyan, S. (1986) Worst case bound for an LRF schedule for the mean weighted flow-time problem, SIAM J Comput., 15, 1119-1129.

14.Klimov, G.P. (1974) Time-sharing service systems, I. Theory Probab. Appl. 19, 532-551.

15.Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (1989) Sequencing and scheduling: algorithms and complexity. Technical report BS-R8909, Center for Mathematics and Computer Science, Amsterdam, The Netherlands.

16.Lenstra, J. K., A.H.G. Rinnooy-Kan, P. Brucker (1977) Complexity of machine scheduling problems, Ann. Discrete Math. 1, 343-362.

17.McNaughton, R. (1959) Scheduling with deadlines and loss functions, Management Science 6, 1-12.

18.Meilijson, I., G. Weiss, (1977) Multiple feedback at a single server station. Stochastic Processes Appl. 5, 195-205.

19.Pinedo, M. (1983) Stochastic scheduling with release dates and due dates, Operations Research 31, 559-572.

20.Pinedo, M., G. Weiss (1987) The largest variance first policy in some stochastic scheduling problems, Oper. Res. 35, 5, Sept.-Oct.

21.Sevcik K.C. (1974) Scheduling for minimum total loss using service time distributions. J. Assoc. Comp. Mach. 21, 66-75.

22.Shapiro, J.F. (1968) Turnpike planning horizons for a Markovian decision model, Management Science 14, 292-300.

23.Smith, W.E. (1956) Various optimizers for single stage production, Naval Research Logistics Quarterly, 3, 59-66.

24.Weber, R. R., (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time, J. Appl. Probab., 19, 167-182.

25.Weber, R. R., P. Varaiya and J. Walrand, (1986) Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime, J. Appl. Probab. 23, 841-847.

26.Weber R.R., Weiss, G. (1990) On an index policy for restless bandits, J. Applied Probability to appear.

27.Wein, L.M. (1987) Asymptotically optimal scheduling of a two station multiclass queueing network. PhD Dissertation, Stanford University.

28.Weiss, G. (1988) Branching bandit processes. Probab. Eng. Inform. Sci., 2, 269-278.

29.Weiss, G. (1990) Approximation results in parallel machines stochastic scheduling. Annals of Operations Research Special Volume on Production Planning and Scheduling, M. Queyranne editor. To appear.

30.Weiss, G., C. C. Huang, C. L. Li, S. Liu, M. Pinedo, J. Song (1987) A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime. Georgia Tech ISyE technical report no. J-87-12.

31.Weiss, G. and M. Pinedo, (1980) Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions, J. Appl. Probab., 17, 187-202.

32.Whittle, P. (1988) Restless Bandits: Activity allocation in a changing world. in A Celebration of Applied Probability J. Gani editor, J. appl. Probab. Special Volume 25A, 287-298.

Industrial and Systems Engineering
Report Series No. J-89-07
August, 1989
Revised November, 1989

ON THE OPTIMAL ORDER OF M MACHINES IN TANDEM*

Chueng Chiu Huang, Gideon Weiss.

* Research supported in part by NSF grant ECS-8712798.

ON THE OPTIMAL ORDER OF M MACHINES IN TANDEM*

Chueng Chiu Huang, Gideon Weiss.

Abstract

We consider M machines in tandem with an infinite supply of jobs and no intermediate storage, and look for the order of the machines which will maximize the throughput. We show that if processing times on the machines are comparable in the sense of likelihood ratio, then it is optimal to use slower machines in the first and last position than in the second and penultimate positions. For 3 and 4 machines this implies bowl shape order of the machines.

* Research supported in part by NSF grant ECS-8712798.

1. INTRODUCTION :

We consider a flowshop of M machines in tandem (series), with no intermediate storage between machines. An infinite supply of jobs are available in front of the 1^{st} machine. The amount of processing required by job j on machine i is X_{ij}. We assume that all X_{ij}'s are independent random variables, and that for a fixed i, $i=1,\dots,M$, the processing times of all the jobs, $j=1,2,\dots$ on machine i are identically distributed, that is $X_{ij} \cong X_i$, $i=1,\dots,M$. Because there is no intermediate storage, blocking of machines can occur - machine i is blocked if job j has completed its processing on machine i, but machine $i+1$ still contains job $j-1$. Blocking as well as processing time determines the throughput of the flowshop - the amount of blocking may depend on the order in which the machines are arranged. The problem is to find the permutation of machines which will maximize the throughput.

This problem has been investigated by several authours, see among others Hillier and Boling (1967), Yamazaki and Sakasegawa (1975), Muth (1984), Pinedo (1982), Yamazaki, Sakasegawa and Shanthikumar (1989). Analitical as well as numerical work has indicated some general principles for ordering the machines: For 3 machines Yamazaki and Sakasegawa (1975) have shown that if the X_i's are comparable in the sense of stochastic ordering then the best order is to put the two slowest machines first and last, and the fastest machine in the middle. This gives rise to the "bowl shape" order (sometimes also called SEPT-LEPT) - arrange machines in order $1,\dots,M$ with

$$E(X_1) \geq E(X_2) \geq \dots \geq E(X_{m-1}) \geq E(X_m) \leq E(X_{m+1}) \leq \dots \leq E(X_M).$$

On the other hand, if there are 3 slow machines (processing time $\simeq \exp(1)$) and a number of very fast machines (processing time $\simeq 0$), the fast machines can be considered as providing finite buffers for storage of jobs, and the optimal solution is to have the slow machines in the first, middle and last position, with the buffers divided equally between them (Pinedo 1982). This gives rise to a "saw tooth" order. It is very difficult to determine in general when one or the other of these rules provides a valid optimal order. Perhaps the principle formulated by Yamazaki et al, "Slow machines should be kept as separate as possible" is the best general rule. Yamazaki, Sakasegawa and Shanthikumar (1989) have carried out extensive numerical work to support this rule.

In the present note we give an elementary and brief proof that if the processing times on the first two and last two machines are comparable in the sense of Likelihood Ratio Ordering then the optimal schedule is to have $X_1 \underset{LR}{\geq} X_2$ and $X_{M-1} \underset{LR}{\leq} X_M$. For 3 and 4 machines this implies bowl shaped order. We note however that for 4 or more machines this does not necessarily imply that the overall slowest two machines are placed in positions 1 and M_{j-1}, we have currently no proof of this stronger conjecture.

Let X and Y denote continuous nonnegative random variables with densities f and g. We say that X is larger then Y in the sense of likelihood ratio ordering, denoted by $X \underset{LR}{\geq} Y$ if

$$\frac{f(x)}{g(x)} \leq \frac{f(y)}{g(y)} \quad \text{for all } x \leq y.$$

Recently Shanthikumar and Yamazaki (1989) have derived results similar to ours, using different methods.

2. FORMULATION AND NOTATION :

The j^{th} cycle of machine i consists of a positive processing period X_{ij}, followed by a nonnegative blocking period B_{ij}, followed by a nonnegative idle period I_{ij}; we call $H_{ij} = X_{ij} + B_{ij}$ the ij^{th} holding period; we call $C_{ij} = X_{ij} + B_{ij} + I_{ij}$ the ij^{th} cycle. Note that machine 1 is never idle, machine M is never blocked. In what follows we use for convenience the index j to label the cycles on machines $1,\dots,M$, which overlap the period H_{1j}. To make this precise if $1j$ labels the processing of job j' on the 1^{st} machine then ij labels the processing of job $j'-i+1$ on the i^{th} machine. We shall name the ij^{th} processing cycle job J_{ij}. Figure 1 illustrates processing on 4 machines.

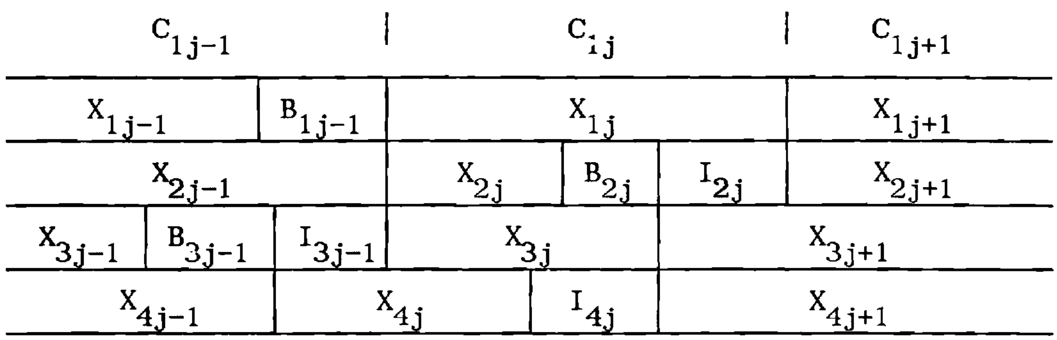


Figure 1 : 4 machines flowshop.

Let the start time of J_{11} be $S_{11} = 0$. We now define several quantitives which enable us to express starting times of the jobs recursively, we denote the start of job J_{ij} by S_{ij}. Define D_{ij} for $i = 1,\dots,M$, $j = 0,1,\dots$ by :

$$S_{ij} = S_{1j} - D_{ij-1} \qquad (2.1)$$

Note that $D_{1j} = 0$, $D_{2j} = 0$, and, by the labelling convention, $D_{ij} \geq 0$.

Let $R_{ij} = X_{ij} - D_{ij-1}$, then job J_{ij} is completed at time

$$S_{1j} + R_{ij} = S_{ij} + X_{ij} \qquad (2.2)$$

Let $G_{ij} = \max_{i-1 \le k \le M} \{R_{kj}\}$, $i = 2,\dots,M$, $G_{1j} = G_{2j}$ then job J_{i-1j} is held, by its own processing or by being blocked until time

$$S_{1j} + G_{ij} = \max_{i-1 \le k \le M} \{S_{kj} + X_{kj}\}. \qquad (2.3)$$

We then have

$$S_{ij+1} = S_{1j} + G_{ij} \qquad \text{for } i = 1,\dots,M \qquad (2.4)$$

$$= \max_{i' \le k \le M} \{S_{kj} + X_{kj}\} \qquad i' = \max\{1, i-1\} \qquad (2.5)$$

In particular we see by comparing (2.4) and (2.1), (with $j+1$ substituted for j in (2.1)), that

$$D_{ij} = G_{1j} - G_{ij}.$$

We also note that machine i will be idle following departure of job J_{ij} for the duration

$$I_{ij} = D_{i+1j} - D_{ij}.$$

Note that : $\qquad D_{i+1j} \ge D_{ij} \qquad i = 1,\dots,M-1.$

In addition to Likelihood ratio ordering we shall also use the more common relation of stochastic ordering. We say $X \underset{ST}{\ge} Y$ if $P(X > a) \ge P(Y > a)$ for all a, or equivalently, if $E[h(X)] \ge E[h(Y)]$ for all nondecreasing functions h. We note that $X \underset{LR}{\ge} Y$ implies $X \underset{LR}{\ge} Y$. In what follows we will require the following lemma :

<u>LEMMA 1</u> (Brown and Solomon 1973) :

Suppose X, Y are independent and $X \underset{LR}{\ge} Y$. If h is a real valued function satisfying $h(x,y) \ge h(y,x)$ whenever $x \ge y$ then $h(X,Y) \underset{ST}{\ge} h(Y,X)$.

3. MAIN RESULT :

We will first investigate the effect of pairwise exchange in the processing times of J_{M-11} and J_{M1} on the starting times of all the succeeding jobs. We condition on the values of D_{i0}, $i = 1,\dots,M$ and X_{ij} for all pairs ij except X_{M-11}, X_{M1}. Let $S_{ij}(x,y)$ be S_{ij} when $X_{M-11} = x$, $X_{M1} = y$. We now compare $S_{ij}(x,y)$ with $S_{ij}(y,x)$. Note that the following Proposition deals with deterministic, non random quantities.

PROPOSITION 1 :

If $x > y$ then $S_{ij}(x,y) \geq S_{ij}(y,x)$.

Proof :

Substitute $j = 1$ in (2.5) to get

$$S_{i2}(x,y) = \max\{\max_{i'\leq k\leq M-2}\{X_{k1}-D_{k0}\}, x-D_{M-10}, y-D_{M0}\} \qquad i = 1,\dots,M$$

$$S_{i2}(y,x) = \max\{\max_{i'\leq k\leq M-2}\{X_{k1}-D_{k0}\}, y-D_{M-10}, x-D_{M0}\} \qquad i' = \max\{1, i-1\}$$

as defined, $D_{ij} \leq D_{i+1j}$. Hence $S_{i2}(x,y) \geq S_{i2}(y,x)$. We proceed by induction, suppose $S_{ij}(x,y) \geq S_{ij}(y,x)$ for $i = 1,\dots,M$ then

$$S_{ij+1}(x,y) = \max_{i'\leq k\leq M}\{X_{kj} + S_{k-1j}(x,y)\}$$

$$S_{ij+1}(y,x) = \max_{i'\leq k\leq M}\{X_{kj} + S_{k-1j}(y,x)\}$$

implies $S_{ij+1}(x,y) \geq S_{ij+1}(y,x)$. □

We now consider the implications of the proposition on for stochastic processing times. Let $X \underset{LR}{\geq} Y$, then by Lemma 1,

$$S_{ij}(X,Y) \underset{ST}{\geq} S_{ij}(Y,X). \qquad (3.1)$$

Since (3.1) holds conditionally for any fixed values of D_{i0} and X_{ij}, $ij \neq M\text{-}11, M1$, it also holds unconditionally.

If for some j_0, we choose $X_{M-1j_0} \sim X$, and $X_{Mj_0} \sim Y$ then a pairwise exchange will result in S_{ij} which is stochastically smaller, for all $j > j_0$. Hence, S_{ij} is stochastically minimized if we choose $X_{M-1j} \sim Y$, $X_{Mj} \sim X$ for all j. We therefore have:

PROPOSITION 2:

Assume $X \underset{LR}{\geq} Y$. Let schedule A have $X_{M-1j} \sim Y$, and $X_{Mj} \sim X$. Let schedule B have $X_{M-1j} \sim X$, and $X_{Mj} \sim Y$. Let schedules A and B be the same for all other jobs. Then for $i = 1,\ldots M$, $j = 1,2,\ldots$ we have: $S^A_{ij} \underset{ST}{\leq} S^B_{ij}$.

Proof: Consider a problem in which we have the freedom to choose separately for each j between two possiblities: Either have $X_{M-1j} \sim Y$, $X_{Mj} \sim X$, or have $X_{M-1j} \sim X$, $X_{Mj} \sim Y$. Let Π be an arbitrary schedule for this problem. Then we can change Π by a series of pairwise exchanges of X_{M-1j} and X_{Mj} to schedule A in such a way as to stochastically decrease all S_{ij}'s at each exchange, and at the same time we can change Π into schedule B in such a way as to stochastically increase all S_{ij}'s. The proposition follows. □

We now want to discuss the effect of using schedule A or B on throughput. There are several ways to define throughput in our context. One general approach is to define the stochastic throughput function

$$\lambda_{in} = \frac{n}{S_{in+1}} \qquad \text{for } i = 1,\ldots,M. \tag{3.2}$$

Here λ_{in} is the (random) number of jobs through machine i per unit time, measured up to the time of start of the $n+1^{st}$ cycle on machine i. It is easy to see that

$$\lambda_{in} \geq \lambda_{1n} \geq \frac{n-i+1}{n} \lambda_{in+i-1} \tag{3.3}$$

Clearly if a steady state throughput is reached as $n \longrightarrow \infty$, then all the

λ_{in}'s will converge to it so they are all asymptotically equivalent. By proposition we have:

$$\lambda_{in}^{A} \underset{ST}{\geq} \lambda_{in}^{B}. \qquad (3.4)$$

We can now state:

THEOREM 1 :

If X_1, X_2 are comparable in the sense of likelihood ratio, and if X_{M-1}, X_M are comparable in the sense of likelihood ratio, then the schedule which achieves the highest steady state throughput will have machines 1,2 and machines M-1,M ordered so that: $X_1 \underset{LR}{\geq} X_2$ and $X_M \underset{LR}{\geq} X_{M-1}$

Proof : As $n \longrightarrow \infty$ for any permutation of the machines there will exist a limiting value of the throughput. By (3.4) the optimal order of the machines will have $X_M \underset{LR}{\geq} X_{M-1}$. By the reversibility property of flowshops (Yamazaki and Sakasegawa (1975), Muth (1979)) the limiting throughput is the same if the order of all the machines is reversed. Therefore the optimal will have $X_1 \underset{LR}{\geq} X_2$. □

For the case of 3 and 4 machines Theorem 1 states that "Bowl Shape" ordering of the machines maximizes throughput.

4. DISCUSSION :

(a) The above results can be generalized somewhat by introducing stochastic order relation between vectors of random variables (see Ross 1983). We say the vector $\underline{X}$ is jointly stochastically larger than the vector $\underline{Y}$, $\underline{X} \underset{ST}{\geq} \underline{Y}$, if for all nondecreasing functions h, $E(h(\underline{X})) \geq E(h(\underline{Y}))$. We note that this definition implies that if $\underline{H}(\underline{X})$ is a vector valued function, such that $\underline{H}(\underline{X})$ is componentwise increasing in $\underline{X}$, then $\underline{X} \underset{ST}{\geq} \underline{Y}$ implies $\underline{H}(\underline{X}) \underset{ST}{\geq} \underline{H}(\underline{Y})$.

It is straight forward to show that the lemma of Brown and Solomon (Lemma 2) extends as follows :

If $\underline{H}(x,y) \geq \underline{H}(y,x)$ componentwise for $x \geq y$, and if X,Y are independent univariate random variables such that $X \underset{LR}{\geq} Y$ then $\underline{H}(X,Y) \underset{ST}{\geq} \underline{H}(Y,X)$.

Using these wider definition we see that proposition 1 implies a strengthened version of (3.1), namely

$$\underline{S}(X,Y) \underset{ST}{\geq} \underline{S}(Y,X)$$

For any vector of job starting times, say

$$\underline{S} = (S_{11},\ldots,S_{M1},S_{12},\ldots,S_{M2},\ldots,S_{1n},\ldots,S_{Mn}).$$

Extending the definition further (Ross 1983), we say that a parametric family (stochastic process) $\{X_\alpha\}_{\alpha \in A} \underset{ST}{\geq} \{Y_\alpha\}_{\alpha \in A}$ if $X_{i_1},\ldots,X_{i_n} \underset{ST}{\geq} Y_{i_1},\ldots,Y_{i_n}$ for any finite subset $\{i_1,\ldots,i_n\} \subseteq A$. Then we have that if $X \underset{LR}{\geq} Y$ then :

$$\{S_{ij}(X,Y)\}_{\substack{i=1,\ldots,M \\ j=1,2,\ldots}} \underset{ST}{\geq} \{S_{ij}(Y,X)\}_{\substack{i=1,\ldots,M \\ j=1,2,\ldots}}.$$

(b) The above extension immediately yields that the throughput functions λ_{in} are jointly stochasticaly ordered, that is (3.4) can be written

$$\{\lambda^A_{in}\}_{\substack{i=1,\ldots,M \\ j=1,2,\ldots}} \underset{ST}{\geq} \{\lambda^B_{in}\}_{\substack{i=1,\ldots,M \\ j=1,2,\ldots}}.$$

Furthermore, we can now introduce an alternative type of throughput function :

$$\lambda_i(t) = \sup\{j \,|\, S_{ij}+X_{ij} \leq t\} \;/\; t$$

$\lambda_i(t)$ is the number of jobs per unit time that were completed on machine i up to time t. Since all $\lambda_i(t)$ are monotone functions of the S_{ij}'s, we have

$$\{\lambda_i^A(t)\}_{\substack{i=1,\dots,M \\ t>0}} \underset{ST}{\geq} \{\lambda_i^B(t)\}_{\substack{i=1,\dots,M \\ t>0}}.$$

(c) We have not made explicit use of the assumption that $X_{ij} \cong X_i$ are i.i.d.. Thus at first glance it may seem that it is enough to require for theorem 1 that $X_{1j} \underset{LR}{\geq} X_{2j}$ for each job j=1,2,..., without assuming $X_{1j} \cong X_1$, $X_{2j} \cong X_2$ (and similarly for M-1,M). This however is not the case — our proof is based on a pairwise exchange argument between X_{M-1j} and X_{Mj-1}, that is we exchange processing times of two different jobs. It is easy to see by counter examples that the assumption of identically distributed processing times on the various machines cannot be totally relaxed.

ACKNOWLEDGEMENT : We wish to thank George Shanthikumar for valueable comments on this paper.

REFRENCES :

Brown M., Sclomon H. (1973) Optimal issuing policies under stochastic field lives. Journal of applied probability 10, pp 761-768.

Hillier F.S., Boling R.W. (1967) Finite queues in series with exponential or erlang service times— a numerical approach. Operations Research 15, pp 286-303.

Muth E.J. (1979) The reversibility property of production lines. Management Science 25, pp. 152-158.

Muth E.J. (1984) Stochastic processes and their network representations associated with a production line queuing model. European J. Operations Research 15, pp 63-83.

Pinedo M., (1982) Minimizing the expected makespan in stochastic flow shops. Operations Research 30, pp 148-162.

Ross S.M., (1983) Stochastic Processes. Wiley, New York.

Shanthikumar J.G., Yamazaki G., personal conmunication.

Yamazaki G., Sakasegawa H., (1975) Properties of duality in tandem queueing systems. Annals Institute Mathematics Statistics 27 , pp 201-212.

Yamazaki G., Sakasegawa H., Shanthikumar J.G., (1989) On optimal arrangement of servers in tandem queues with blocking. paper presented at TIMS/OSAKA International Meeting, July 89.

Industrial and Systems Engineering
Report Series No. J-89-4
July, 1989

ON AN INDEX POLICY FOR RESTLESS BANDITS*

Richard Weber, University of Cambridge
Gideon Weiss, Georgia Institute of Technology

* Research supported in part by NSF grant ECS-8712798.

ON AN INDEX POLICY FOR RESTLESS BANDITS

RICHARD R. WEBER*, *University of Cambridge*
GIDEON WEISS**, *Georgia Institute of Technology*

Abstract

We investigate the optimal allocation of effort to a collection of n projects. The projects are "restless" in that the state of a project evolves in time whether or not it is allocated effort. The evolution of the state of each project follows a Markov rule, but transitions and rewards depend on whether or not the project receives effort. The objective is to maximize the expected time-average reward under a constraint that exactly m of the n projects receive effort at any one time. We show that as m and n tend to infinity with m/n fixed, the per-project reward of the optimal policy is asymptotically the same as that achieved by a policy which operates under the relaxed constraint that an average of m projects be active. The relaxed constraint was considered by Whittle (1988) who described how to use a Lagrangian multiplier approach to assign indices to the projects. He conjectured that the policy of allocating effort to the m projects of greatest index is asymptotically optimal as m and n tend to infinity. We show that the conjecture is true if the differential equation describing the fluid approximation to the index policy has a globally stable equilibrium point. This need not be the case, and we present an example for which the index policy is not asymptotically optimal. However, numerical work suggests that such counterexamples are extremely rare and that the size of the suboptimality which one might expect is minuscule.

FLUID APPROXIMATIONS; GITTINS INDEX; LARGE DEVIATION THEORY; MULTI-ARMED BANDIT PROBLEM; STOCHASTIC SCHEDULING.

* Postal address: Cambridge University Engineering Department, Management Studies Group, Mill Lane, Cambridge CB2 1RX, U.K.

** Postal address: School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0205, U.S.A.

1. Restless Bandits

Whittle (1988) has recently studied an interesting generalization of the classical multi-armed bandit problem. The classical problem concerns n projects, the state of project i at time t being denoted $x_i(t)$. At each time t just one project is to be operated. If project i is operated then a reward $g_i(x_i(t))$ is received and the transition $x_i(t) \to x_i(t+1)$ follows a Markov rule specific to project i. The $n-1$ projects which are not operated produce no reward and their states do not change. One thinks of a gambler who at each turn can pull exactly one of the n arms of a multi-armed bandit, or slot-machine, and who desires to maximize his time-average reward by an optimal sequence of pulls..

Gittins (1970) (see Gittins and Jones (1974)) showed that an *index policy* is optimal for this problem. The *Gittins' index*, denoted by $\nu_i(x_i)$, can be calculated for each project as a function of the label i and state x_i alone ; the optimal policy is simply to operate the project of greatest index.

Whittle has studied a variation in which two generalizations are introduced. Firstly, at each moment exactly m of the projects are to be operated. Secondly, those $n-m$ which are not operated may nevertheless contribute reward and change their state. A project is said to be *active* or *passive* depending upon whether or not it is operated. It is because passive projects may change state that they are called *restless bandits* (since we are now thinking of a multi-armed bandit machine for which even those arms which are not pulled change state). Whittle gave several examples of problems which are nicely modelled as restless bandits, for example the exhaustion and recuperation modes of n workers, where m must always be active. Here a change of state corresponds to a change of a worker's physical condition. A change of state may also correspond to a change of information. One might gain different information when making a project active than when making it passive. For example, projects might deteriorate in an unobserved manner when not made active. We might think of m helicopters trying to keep track of the positions of n submarines.

For simplicity of exposition we suppose that all projects are of the same type: they have the same finite state space, with states labelled $\{1, \ldots, k\}$, and change their state according to an identical Markov rule. It is convenient to formulate the model in continuous time. We suppose that following entry to state i the next *potential* change of state occurs at a time which is exponentially distributed with mean 1. At that time the project moves to state j with probability $P_{ij}(a)$, where action $a=1$ or 2 denote respectively that the active or passive action was being applied to the state just prior to its potential change of state. Note that the diagonal entries of $P(a)$ are not necessarily 0, so the project may not actually change state. This *uniformization device* is a standard idea in the study of Markov processes and simplifies the analysis. We shall also suppose that the transition matrices are such that the states form a single closed class regardless of the policy employed. Reward is earned at a rate $g(i,a)$ whenever the project is in state i and action a is taken.

If one were attempting to maximize average reward over the infinite horizon for a single project the solution would be found from the optimality equation

$$\gamma + f(i) = \max_{a=1,2} \{ g(i,a) + \sum_{j=1}^{k} P_{ij}(a)f(j) \},$$

and γ would be the time-average optimal reward. Consider now the optimality equation obtained if the reward received when taking the passive action is subsidised by an amount ν.

$$(1) \quad \gamma(\nu) + f(i) = \max_a \{ g(i,1) + \sum_{j=1}^{k} P_{ij}(1)f(j), \quad \nu + g(i,2) + \sum_{j=1}^{k} P_{ij}(2)f(j) \}$$

One thinks of ν as a subsidy which is paid for taking the passive action. It is intuitive that as $\nu \to -\infty$ it is optimal to make all projects active and as $\nu \to +\infty$ it is optimal to make all projects passive. The proportion of time for which it is optimal to take the passive action increases in ν. One can interpret ν as the Lagrangian multiplier associated with a constraint that the passive action be taken for a proportion of the time $1-\alpha$, $0 \leq \alpha \leq 1$. This leads us to consider a problem in which the constraint is relaxed to demanding only that the number of projects to be made active satisfies $Em(t)=\alpha n$ in time average. It is clear that the policy which maximizes the average reward subject to this constraint satisfies (1) when ν is pitched at the right level. A *subsidy policy* is defined by Whittle as a policy induced by (1) for some value of ν and which resolves any tie between the terms within the maximization operator by choosing the passive action. However, except for a finite number of values of α, for which the non-randomizing ν-subsidy policies are optimal, the appropriate subsidy ν will be such that the active and passive actions are equally attractive for some state i. By appropriately randomizing the choice of action in this state a stationary policy is obtained for which the constraint is satisfied. Denote by $r(\alpha)$ the maximum average reward that can be achieved under the relaxed constraint; this can be expressed as (Whittle, Proposition 1)

$$(2) \qquad r(\alpha) = \inf_{\nu} \{ \gamma(\nu) - \nu(1-\alpha) \}.$$

Clearly $nr(\alpha)$ is an upper bound for $R_{opt}^{(n)}(\alpha)$, the maximal average reward to be obtained from n projects subject to the more demanding constraint that exactly $m=\alpha n$ are to be made active at all times. The policy which achieves the value $nr(\alpha)$ simply applies the same policy to each project independently, making each project active, passive, or perhaps randomizing between active and passive actions, on the basis of the state of the project alone. We call this policy, which is optimal under the relaxed constraint, the *relaxed-constraint optimal policy*, or more briefly the *relaxed policy*, and denote it σ_{rel}.

[*Remark.* If α and n are such that αn is not an integer, then we interpret the constraint $m=\alpha n$ as demanding that $\lfloor \alpha n \rfloor$ projects be made active, $n-\lceil \alpha n \rceil$ projects be made passive, and the remaining project be made active with probability $\alpha n-\lfloor \alpha n \rfloor$. This is consistent with the idea that the resource available for applying active actions is continuously divisible. This interpretation also avoids technical issues that are not central to our discussion.]

Whittle defines the index $\nu(i)$ for a project in state i as the least value of the subsidy ν for which it could be optimal in (1) to make the project passive in state i. It turns out that this index reduces to the usual Gittins' index in the case that the passive projects do not change state and yield no rewards. If indexing is to be meaningful it should induce a consistent ordering, meaning that if it is optimal to make a project passive when the subsidy is ν then it is also optimal to make it passive for all $\nu' > \nu$. The concept is formalized in Whittle's definition of *indexability.*

Definition. Let $D(\nu)$ be the set of states for which a project would be made passive under a ν-subsidy policy. The project is indexable *if $D(\nu)$ increases monotonically from $\emptyset$ to $\{1, \ldots, k\}$ as ν increases from $-\infty$ to $+\infty$.*

Whittle's *index policy,* denoted σ_{ind}, is the one which at all times makes active the m $(=\alpha n)$ projects of greatest index (where this is interpreted according to the remark above). For this policy we denote the time-average reward for a problem with n projects by $R_{ind}^{(n)}(\alpha)$. Clearly

(3) $$R_{ind}^{(n)}(\alpha) \leq R_{opt}^{(n)}(\alpha) \leq nr(\alpha).$$

One expects that under the relaxed policy the optimal reward per project will be close to $r(\alpha)$ and the policy will be nearly optimal. This is Whittle's conjecture, namely

Conjecture. (Whittle (1988))

(4) $$R_{ind}^{(n)}(\alpha)/n \rightarrow r(\alpha) \text{ as } n,\ m \rightarrow \infty,\ \ m=\alpha n.$$

Because it is possible to compute the $\nu(i)$'s the index policy is easy to implement. The truth of the conjecture would make the index policy an attractive policy for the constrained problem, since we could be sure that for large n it nearly achieves a reward of $nr(\alpha)$; this is also a quantity that can be computed.

The conjecture is plausible since, as n increases, one expects a weaker coupling between the states of distinct projects. If the relaxed policy is applied to n projects then the equilibrium number in state i will be binomially distributed as $B(n\pi_i\,,\ n\pi_i(1-\pi_i))$, where π_i is the proportion of time a single project spends in state i. If the initial distribution is the relaxed policy's equilibrium distribution and one starts to apply the index policy then, at least initially, the relaxed and index policies will differ in the actions they take on a number of states whose expected value is only $O(\sqrt{n})$, and the expected difference in reward per project between the policies will be $O(1/\sqrt{n})$. Since actions don't differ very much the equilibrium distribution of the index policy will also have about $n\pi_i \pm O(\sqrt{n})$ projects in state i.

However, we have discovered that conjecture (4) is false and we show this in section 4 using ideas from the theory of large deviations and a specific counterexample. The counterexample was by no means easy to find. We still believe the conjecture to be true in most circumstances. Section 3 describes an analysis of the index policy using the theory of large deviations. In it we give a sufficient condition for the truth of (4). Section 2 begins the paper with a proof of a positive result: that the second inequality in (3) is an equality to within $O(\sqrt{n})$. Thus for large n imposition of the more demanding constraint does not lead to substantial reduction of reward per project.

2. The asymptotic reward of the optimal policy

We begin by showing that asymptotically nothing is gained by the relaxation of the constraint. Asymptotically the optimal reward per project is the same for the constrained and relaxed problems.

Theorem 1.

$$(5)\qquad R_{opt}^{(n)}(\alpha)/n \to r(\alpha) \quad as\ n,\ m \to \infty,\ \ m=\alpha n.$$

Proof. Let π be the equilibrium distribution for the state of a single project when the relaxed policy is employed. Consider the expected-average-cost optimality equation for the constrained problem.

$$R_{opt}^{(n)}(\alpha)/n + f(x) \;=\; \max_a \{ (1/n) \sum_{i=1}^{n} g(x_i\,,\ a_i) \;+\; E_a f(X) \;\}.$$

Here x_i denotes the state of project i, a_i denotes the action taken for that project, and X_i denotes the state of project i subsequent to the next potential transition (which occurs after time exponentially distributed with mean $1/n$), x, $X \in \{0,\ldots,k\}^n$, $a \in \{1,2\}^n$. Assume data in the problem is such that π is rational and n is such that $n\pi$ is vector of integers. Let $n_i(x)$ denote the number of projects in state i. Suppose the state x is such that the number of projects in state i is exactly $n_i(x)=n\pi_i$ Then, application of the relaxed policy satisfies the constraint that exactly m projects will be made active. Suppose now that the relaxed policy is applied to every project; denote by a_i^* the action applied to project i. Moreover, suppose that for a time δ the constant action a_i^* is applied to project i even if that project changes state. The expected number of potential state changes which occur during this interval of length δ is $n\delta$. The expected reward obtained during the interval is bounded below by $\delta r(\alpha)n - n\delta^2 G$, where $G=2max_{i,a}|\ g(i,a)\ |$. Clearly the policy is suboptimal, so

$$\text{(6)} \qquad \delta R_{opt}^{(n)}(\alpha) + f(x) \ \geq \{\ \delta r(\alpha)n - n\delta^2 G + E_{a^*}f(X^\delta)\ \}.$$

Here X_i^δ is the state of the project i after time δ. Define, for any two states x and y, the distance $d(x,y)$ as the minimal number of components in which x and y can be made to differ by permuting the components of y: this is $d(x,y)=(1/2)\sum_i|\ n_i(x)-n_i(y)\ |$. Note that $n_i(X^\delta)=Y_1+\cdots+Y_k$ where Y_j has a binomial distribution $B(n_j(x), P_{ji}(a_j,\delta))$ and $P_{ji}(a_j,\delta)=e^{-\delta}(I+\delta P_{ji}(a_j)+\delta^2 P_{ji}^2(a_j)/2!+\cdots)$ denotes the probability with which a project in state j is in state i after time δ given that the fixed action a_j is applied. From this, and the fact that $n_i(x)=n\pi_i$, it follows that the expected value and variance of X_i^δ are $n_i(x)+no(\delta)$ and $-2\delta\, n_i\{1-P_{ii}(a_j)\}+no(\delta)$. By the central limit theorem and the fact that $d(\cdot)$ satisfies a triangle inequality, it follows that $Ed(x,X^\delta)\leq no(\delta)+A\sqrt{n\delta}$ for some A.

Suppose we could show that there exists a $B>0$ such that $f(y)-f(x)\ \geq\ -Bd(x,y)$, for any x and y. Then from (6) we would have

$$\text{(7)} \qquad r(\alpha)\ \geq\ R_{opt}^{(n)}(\alpha)/n\ \ \geq\ \ r(\alpha) - \delta G - B\{\,o(\delta)/\delta + A/\sqrt{n\delta}\,\}.$$

By taking δ sufficiently small and then letting $n\to\infty$ (through a subsequence for which $n\pi\in Z^k$) the right hand side of (7) has a limit greater that $r(\alpha)-\epsilon$, for any $\epsilon>0$. This would prove the theorem.

Since $d(\cdot)$ obeys a triangle inequality, it suffices to prove the claim of the above paragraph to consider x, y such that $d(x,y)=1$. Suppose this is the case and that x and y differ on project i. We use a coupling argument: suppose that in state y we apply to each project exactly the same action which σ_{opt} applies to that project in state x. We continue to do this until there is a potential change of state for project i. This occurs after a time which is exponentially distributed with mean 1 and we deduce from the optimality equation

$$\text{(8)} \qquad f(x) - f(y) \geq\ \ -G\ +\ \ E_a\{\ f(X)-f(Y)\},$$

where X and Y denote the states of the projects at the time of the first potential change of state for project i. Now $d(X,\ Y)\leq 1$. Moreover, there is always some probability, say at least $\theta>0$, that $X_i=Y_i$ and therefore that $d(X,Y)=0$. Thus from (8) , we have

$$\min_{x,y:\ d(x,y)=1}\{ f(x) - f(y) \} \ \geq\ -G\ +\ (1-\theta) \min_{x,y:\ d(x,y)=1} \{ f(x)-f(y) \},$$

which implies $f(x)-f(y)) \geq -G/\theta$, completing the proof of the theorem.

3. A sufficient condition for asymptotic optimality of the index policy

In this section we consider the index policy, σ_{ind}, applied to a collection of n projects. Let $z_n(t)=(z_{n1}(t), \ldots, z_{nk}(t))$ be a state for the system, where $z_{ni}(t)$ denotes the proportion of projects in state i. Possible transitions of the process are of the form $z_n \to z_n+(1/n)e_{ij}$, where e_{ij} is a vector with -1 in the i'th component, $+1$ in the j'th component and 0's in all other components. Consider a policy in which the $m=\alpha n$ projects of greatest index are made active. We extend the definition of the index policy by stating that when αn is not an integer then $\lfloor \alpha n \rfloor$ projects of greatest index are made active and then one further project, with a greatest index amongst those remaining, is made active with probability $\alpha n-\lfloor \alpha n \rfloor$. Let q^1_{ji} and q^2_{ji} denote the transition rates from state i to j under the active and passive actions respectively. For convenience, we have chosen to write the transition rate matrices so that (q^1_{ji}), and (q^2_{ji}) have columns summing to zero (which is contrary to the usual convention for Markov processes, but convenient in this exposition). The remainder of the paper does not employ the uniformization, but works directly with the transition rates. Define for any numbers a^1 and a^2, and $1 \leq i \leq k$,

$$u_i(z) = \min\Big\{ z_i\ ,\ \max\ \{\ 0\ ,\ \alpha - \sum_{h=i+1}^{k} z_h\ \}\Big\}/z_i$$

$$1-u_i(z) = \min\Big\{ z_i\ ,\ \max\ \{\ 0\ ,\ 1-\alpha - \sum_{h=1}^{i-1} z_h\ \}\Big\}/z_i.$$

$$\phi_i(z,\ a^1,\ a^2) = u_i(z)a^1 + (1-u_i(z))a^2$$

Here $u_i(z)$ and $1-u_i(z)$ are the probabilities that a project which is selected at random from amongst those which are presently in state i will be made active or passive. Under the index policy the transition rate associated with e_{ij} is $nq_{ji}(z_n)z_{ni}$, where

(9) $$q_{ji}(z) = \ \phi_i(z,\ q^1_{ji},\ q^2_{ji}) = \{\ u_i(z)q^1_{ji} + (1-u_i(z))\ q^2_{ji}\ \}.$$

Define the path $z(t)$ starting at $z(0)$, $\sum z_i(0)=1$, by the differential equation

(10) $$dz/dt = \sum_{i,j} q_{ji}(z)z_i e_{ij} = Q(z)z,$$

or equivalently, $dz_i/dt = \sum_{j \neq i} q_{ij}(z)z_j - \sum_{j \neq i} q_{ji}(z)z_i$. This is the *fluid approximation* for $z_n(t)$.

A sufficient condition for the asymptotic optimality of the index policy can be stated as follows.

Theorem 2. Let π be the equilibrium distribution of a single project operated under the relaxed-constraint optimal policy. Suppose that the differential equation (10) *has no limit cycles, nor do its solutions behave chaotically. Then if the projects are indexable,* (10) *has the unique fixed point π and $z(t) \to \pi$ for all $z(0)$. Furthermore conjecture* (4) *is true: $R^{(n)}_{ind}(\alpha)/n \to r(\alpha)$,* as n and m increase to infinity with $m = \alpha n$.

Firstly, we prove a lemma which shows that indexability implies that π is a unique fixed point. Consider a single project. Suppose it is indexable and that, without loss of generality, $\nu(1) < \cdots < \nu(k)$. Let $\sigma(i,\theta)$ be the policy which takes the passive action in states 1, ..., $i-1$, the active action in states $i+1$, $\cdots$, k, and which takes the passive and active actions in state k with probabilities θ and $1-\theta$ respectively, $0 \leq \theta \leq 1$. Let $\alpha(i,\theta)$ denote the time-average proportion of time that the active action is taken using policy $\sigma(i,\theta)$.

Lemma 1. Suppose the project is strictly indexable, such that $\nu(1) < \cdots < \nu(k)$. Then $\alpha(i,\theta)$ is a strictly decreasing function of θ.

Proof. $\gamma(\nu)$ is a convex, piecewise-linear, increasing function of ν. This follows from the fact that the in the set S of stationary nonrandomizing policies, a policy $\sigma \in S$ has a reward function, say $\gamma_\sigma(\nu)$, which is linear in ν and $\gamma(\nu) = \max_{\sigma \in S} \{ \gamma_\sigma(\nu) \}$. Also, $\gamma(\nu)$ is strictly increasing for $\nu > \nu(1)$. By indexability and the definitions of $\nu(i-1)$ and $\nu(i)$ we have that

$$\gamma_{\sigma(i,0)}(\nu) = \gamma(\nu(i-1)) + \Big(\nu - \nu(i-1)\Big)\Big(1 - \alpha(i,0)\Big) \text{ and}$$

$$\gamma_{\sigma(i,1)}(\nu) = \gamma(\nu(i)) + \Big(\nu - \nu(i)\Big)\Big(1 - \alpha(i,1)\Big)$$

are both subgradients to $\gamma(\nu)$ at $\nu = \nu(i)$. Hence

$$(11) \qquad \gamma(\nu(i)) = \gamma(\nu(i-1)) + \Big(\nu(i) - \nu(i-1)\Big)\Big(1 - \alpha(i,0)\Big)$$

Now since $\nu(i-1) < \nu(i)$,

$$(12) \qquad \gamma_{\sigma(i,1)}(\nu(i-1)) = \gamma(\nu(i)) + \Big(\nu(i-1) - \nu(i)\Big)\Big(1 - \alpha(i,1)\Big) < \gamma(\nu(i-1)).$$

So (11) and (12) imply $\alpha(i,0)>\alpha(i,1)$. Now suppose that π^0 and π^1 are the equilibrium distributions of policies $\sigma(i,0)$ and $\sigma(i,1)$. The equilibrium distribution induced by $\sigma(i,\theta)$ is a linear combination of π^0 and π^1, namely $\pi^\theta=(1-\rho)\pi^0 + \rho\pi^1$, where $\rho=\theta\pi_i^0/\{\theta\pi_i^0+(1-\theta)\pi_i^1\}$. Note that $\pi_i^\theta=\pi_i^0\pi_i^1/\{\theta\pi_i^0+(1-\theta)\pi_i^1\}$. Thus $\alpha(i,\theta)=1-(\pi_1^\theta+\cdots+\pi_{i-1}^\theta+\theta\pi_i^\theta)$ is the ratio of two linear functions of θ and is therefore monotone as θ goes from 0 to 1. Since $\alpha(i,0)>\alpha(i,1)$ it must be strictly decreasing. This proves the lemma.

We shall also make use of the following theorem, due to Alan Weiss (1988), which states that on average z_n does not much differ from a fixed distribution ζ.

Proposition. (*see Weiss* (1988) *Theorem* 2) *Suppose there exists a probability distribution* ζ *such that for every initial probability distribution* $z(0)$ *the fluid approximation* $dz/dt=Q(z)z$ *has* $z(t)\to\zeta$, *and the transition rates* $q_{ij}(z)$ *are bounded and Lipschitz-continuous. Then for every* $\epsilon>0$ *there exist positive constants* c_1 *and* c_2 *such that for any initial state* $z_n(0)$

(13)
$$\lim_{t\to\infty}\frac{1}{t}\int_0^t P(\ \|z_n(u)-\zeta\|_2>\epsilon\ )\,du \le c_1e^{-nc_2}.$$

Weiss' proof of this result follows the ideas in Freidlin and Wentzell (1984), where similar theory is developed for diffusions, rather than Markov jump processes. We have stated Weiss' proposition in a form in which we will use it. In fact, we can also have state dependent exogenous arrival and departure rates, provided these are also Lipschitz-continuous.

Proof of theorem 2. Note that our $q_{ij}(z)$ are indeed Lipschitz-continuous. Observe also that in state $z_n(t)$ the index policy has instantaneous reward per project of $\sum_i\phi_i(z_n(t),\ g(i,1),\ g(i,2))$, where we use the function ϕ_i that was defined at the start of this section to evaluate the reward obtained by the mixing of active and passive actions. Similarly, if π is the equilibrium distribution of the state of a single project under the relaxed policy then $r(\alpha)=\sum_i\phi_i(\pi,\ g(i,1),\ g(i,2))$. Clearly $\phi_i(z_n(t),\ g(i,1),\ g(i,2))$ is a continuous function of z over a compact region. Suppose for the moment that (10) has the unique equilibrium π. Since we assume (10) has no limit cycles or chaotic behaviour it must be that for any initial distributions $z(0)$, $z(t)\to\pi$ as $t\to\infty$. Taking $\zeta=\pi$ Weiss' proposition implies that the difference between $r(\alpha)$ and $R_{ind}^{(n)}(\alpha)/n$ can be bounded by

$$2\max_{i,a}|\,g(i,a)\,|\,c_1e^{-nc_2} + \sup_{z:\|z-\pi\|_2<\epsilon}\sum_i|\,\phi_i(z,\ g(i,1),\ g(i,2))-\phi_i(\pi,\ g(i,1),\ g(i,2))\,|\ .$$

This may be made smaller than any arbitrary η, by first choosing ϵ such that the second term is less than $\eta/2$ and then taking n large enough that the first term is also less than $\eta/2$.

The proof of the theorem is completed by showing that $Q(z)z$ has the unique zero $Q(\pi)\pi=0$. Note first that $Q(\pi)\pi=0$, since (from (10) and the following line) this is simply a statement of the balance equations for the equilibrium distribution of the relaxed policy. Similarly, if $\zeta\neq\pi$ were a second probability vector such that $Q(\zeta)\zeta=0$, then ζ would have to be the stationary distribution of some other policy of the form $\sigma(i,\theta)$, such that $\alpha(i,\theta)=\alpha$. But this would contradict lemma 1. Hence $Q(z)z=0$ has the unique root $z=\pi$. This completes the proof of theorem 1.

4. Suboptimality of the index policy

It has been established that conjecture (4) is true if the differential equation for the fluid approximation has an equilibrium point which is globally stable within the $k-1$ dimensional space of probability vectors. Indexability was used as sufficient condition to guarantee uniqueness of the equilibrium point. In this section we begin by showing that even if it were known by some other argument that the equilibrium point is unique, indexability is a necessary condition for the stability of that point. Although lemma 2 is interesting in itself, the main reason for its presentation is in order to explain how the question of the stability of the equilibrium point of (10), which is apparently a nonlinear differential equation, reduces to a question of the stability of a linear system. In the second part of this section we use these ideas to explain a counterexample to (4) that occurs because the equilibrium point is unstable.

Lemma 2. Suppose that for a given α the stable point of (10) *is the equilibrium distribution π, and i is a state such that, $0<u_i(\pi)<1$; $u_j(\pi)=0$, $j<i$; $u_j(\pi)=1$, $j>i$. Suppose that π^0 and π^1 are the equilibrium distributions of policies $\sigma(i,0)$ and $\sigma(i,1)$. Recall that $\sigma(i,0)$ is the nonrandomizing policy which takes the active action in states $i, \dots, k$, and the passive action in states $1, \dots, i-1$. $\sigma(i,1)$ is the nonrandomizing policy which takes the active action in states $i+1, \dots, k$, and the passive action in states $1, \dots, i$. Then in order that $z(t)\to\pi$ for all $z(0)$ it is necessary that $\alpha(i,1)<\alpha(i,0)$. Equivalently this is*

$$(14) \qquad \alpha(i,1) = \pi^1_{i+1}+\cdots+\pi^1_k < \pi^0_i+\cdots+\pi^0_k = \alpha(i,0).$$

Note that (4) might be true for some values of α and untrue for others. Condition (14) must hold if (4) is true for any α between $\alpha(i,1)$ and $\alpha(i,0)$. If (4) is true for all α then indexability is required.

Proof. Let q_j^k be the j'th column of the matrix (q_{ij}^k). In a region C_i, defined as the closure of the set $\{ z: \ 0<u_i(z)<1, \sum z_i=1 \}$, equation (10) can be written

$$dz(t)/dt = A_i z(t) + b, \quad \text{where}$$

$$b = (1-\alpha)\, q_i^2 + \alpha\, q_i^1,$$

$$A_i = (\ q_1^2 - q_i^2 \mid \cdots \mid q_{i-1}^2 - q_i^2 \mid \ 0 \ \mid q_{i+1}^1 - q_i^1 \mid \cdots \mid q_k^1 - q_i^1 \ ), \tag{15}$$

and A_i is a matrix partitioned by columns. Interestingly, (10) in just a linear differential equation in region C_i. Indeed dz/dt is linear in z in each of k regions, $C_1, \ldots, C_k$. We can eliminate z_i from the right hand side of (10). (This is a consequence of the fact that z is constrained to the $k-1$ dimensional subspace of probability vectors.) Let $\tilde{A}_i$ be the $(k-1)\times(k-1)$ matrix formed from A_i by deleting the i'th row and column and let $\tilde{z}$ and $\tilde{\pi}$ be $k-1$ dimensional vectors formed from z and π by deleting the i'th component. From $d(\tilde{z}-\tilde{\pi})/dt=\tilde{A}_i(\tilde{z}-\tilde{\pi})$ we see that if $z(t)\to\pi$ for all $z(0)$ then A_i must be a stability matrix. Thus the eigenvalues of A_i must have negative real parts. Using the fact that

$$-q_i^k = \left(\sum_{j<i} q_j^2 \pi_j^{k-1} + \sum_{j>i} q_j^1 \pi_j^{k-1} \right) / \pi_i^{k-1}, \ k=1,2,$$

we can eliminate q_i^1 and q_i^1 from (15), to get $\tilde{A}_i=\tilde{Q}_i B_i$, where

$$\tilde{Q}_i = (\ q_1^2 \mid \cdots \mid q_{i-1}^2 \mid \ q_{i+1}^1 \mid \cdots \mid \ q_k^1 \ ),$$

$$B_i = I_{k-1} + (\ \pi^1/\pi_i^1 \ \mid \cdots \mid \pi^1/\pi_i^1 \mid \pi^0/\pi_i^0 \mid \cdots \mid \pi^0/\pi_i^0 \ ).$$

Here B_i is the sum of the identity matrix and a rank-2 matrix having all its first $i-1$ and last $k-i$ columns identical. $\tilde{Q}_i$ is a Metzler matrix (of negative diagonal and nonnegative off-diagonal entries) and has nonpositive column sums. Therefore its Perron-Frobenius eigenvalue is nonpositive. In fact, this eigenvalue cannot be zero, for if $\xi>0$ were the corresponding eigenvector we would have $Q(\pi)(\xi_1, \ldots \xi_{i-1}, 0, \xi_{i+1}, \ldots, \xi_k)^T = 0$, which contradicts the irreducibility of $Q(\pi)$. Hence the real parts of the eigenvalues of $\tilde{Q}_i$ must negative and $\tilde{Q}_i$ is a stability matrix. A necessary condition for a $d\times d$ matrix to be a stability matrix is that its determinant have the same sign as $(-1)^d$ (since this has the sign of the product of the real eigenvalues when all are negative). Therefore $\det(B_i)$ is necessarily positive. It is not hard to show that $\det(B_i)=(1-\pi_1^0-\cdots-\pi_{i-1}^0-\pi_{i+1}^1-\cdots-\pi_k^1)/\pi_i^0\pi_i^1$, which is positive if and only if (14) holds. This completes the proof of the lemma.

Consider now the problem described by the following data in which $k=4$. All the following matrix calculations were carried out using the software PC-MATLAB 3.10.

$$(q^1_{ji}) = \begin{bmatrix} -2.5 & 0.0025 & 0 & 1.0 \\ 0.5 & -0.2825 & 0 & 0 \\ 1.0 & 0.28 & -2.0 & 1.0 \\ 1.0 & 0 & 2.0 & -2.0 \end{bmatrix}, \qquad g(j,1) = \begin{bmatrix} 0 \\ 10 \\ 10 \\ 10 \end{bmatrix}$$

$$(q^2_{ji}) = \begin{bmatrix} -2.5 & 0.5 & 0 & 1.0 \\ 0.5 & -56.5 & 0 & 0 \\ 1.0 & 56.0 & -2.0 & 1.0 \\ 1.0 & 0 & 2.0 & -2.0 \end{bmatrix}, \qquad g(j,2) = \begin{bmatrix} 10 \\ 10 \\ 1 \\ 0 \end{bmatrix}.$$

In states 1, 3 and 4 the transition rates are the same for both passive and active actions. In state 2 the passive rates q^2_{2i} are chosen to be 200 times the active ones q^1_{2i}. The reader can check that projects are indexable and that as the subsidy for passivity increases from $-\infty$ through the values -10, 0, 9 and 10, the set of states which ought to be made passive, $D(\nu)$, increases monotonically by the addition of states 1, 2, 3 and 4 in that order. Suppose α is chosen so that the relaxed-constraint optimal policy makes state 1 passive, states 3 and 4 active, and state 2 passive or active with some probabilities θ and $1-\theta$. Then

$$\tilde{A}_2 = \begin{bmatrix} -3.0 & -0.0025 & 0.9975 \\ -55.0 & -2.28 & 0.72 \\ 1.0 & 2.0 & -2.0 \end{bmatrix}$$

$\tilde{A}_2$ is not a stability matrix since its eigenvalues are -7.4037 and 0.0618 $\pm$ 3.9670 i. This leads to a counterexample to conjecture (4). Suppose we take $\alpha=0.835$. For this value of α the relaxed policy is passive on state 1, active on state 3, and active and passive on state 2 with probabilities 916200/925153 (=0.990323) and 8953/925153 (=0.009677) respectively. The equilibrium is $\pi=(0.1644, 0.0973, 0.3281, 0.4102)$.

For this value of α the solution to (10) does not tend to π as $t\to\infty$. Numerical integration of (10) shows that the fluid flow approximation for the index policy actually tends to a limit cycle of period 1.6384498 in which the state for which $0<u_i(z)<1$ alternates between state 1 and state 2. So, in contrast to the relaxed policy, projects in state 1 are sometimes made active. The proof which Weiss gives for the proposition in section 3 can be adapted in an obvious way to a version in which the assumption that $z(t)$ tends to a unique equilibrium, $Q(\zeta)\zeta=0$, is replaced by the assumption that $z(t)$ tends to a unique limit cycle for all initial probability vectors $z(0)$. It can then be shown using arguments similar to those in section 3, that the asymptotic average reward per project of the index policy is the reward obtained by averaging the reward function around the path of the limit cycle. For our data this integral comes to $10-1.26577\text{x}10^{-4}$. Clearly the relaxed policy achieves an average reward of 10. Thus the index policy is asymptotically suboptimal, by the tiny amount of 0.0012%.

5. Conclusions

For the data of the counterexample in section 4 there is a heuristic policy which is asymptotically optimal. Suppose that $m/n=\alpha$ and the relaxed policy takes the active and passive actions in state 2 with probabilities $1-\theta$ and θ. Suppose there are n_2 projects in state 2. Consider the policy which makes $\min\{(1-\theta)n_2,\ m\}$ of these projects active, and then makes enough of the remaining projects in states 1 and 3 active, to bring the total number of active projects to exactly $m=\alpha n$. (Which projects in states 1 and 3 are made active does not matter since the transition rates do not depend on the action taken in these states.) The resulting Markov process is a migration process and satisfies detailed balance equations. So one can find expressions for the equilibrium distribution and show that it has asymptotically the same proportions of projects in each state as the relaxed policy.

Conjecture (4) is always true when $k=2$. In this case the regions C_1 and C_2 have a single point as their boundary. The trajectories of dz/dt must enter one or the other of these regions and never leave it. The only behaviour consistent with this is $z(t)\rightarrow\pi$, so the conditions of theorem 2 are met. In fact, for the case $k=2$ we have derived expressions for the equilibrium distribution of the index policy. One can give a direct proof of the truth of conjecture (4). It turns out that the asymptotic difference between $R_{ind}^{(n)}(\alpha)/n$ and $r(\alpha)$ is even less than $O(1/\sqrt{n})$.

Our counterexample had $k=4$ and it is not clear whether there might a counterexample with $k=3$. Certainly it will be harder to discover such an example, since we can show that when $k=3$ indexability always implies the stability of $\widetilde{A}_i$. Thus the equilibrium point of (10) is at least locally stable.

By randomly generating values for the active and passive matrices, Q^1 and Q^2, we have found a number of counterexamples for $k=4$ and $k=5$. In our earliest experiments we generated the off-diagonal entries in these matrices as uniform random variables in [0,10] and then multiplied the columns by random factors. Roughly 90% of the test problems were indexable, but in a sample of over twenty thousand test problems no counterexample to the conjecture was found. Counterexamples were finally discovered by restricting attention to test matrices for which Q^1 and Q^2 differed in just one column, as in the example of section 4. Our thinking was that for this very specialised case a proof of the conjecture or a counterexample might more easily be discovered. The experiments were rewarded with counterexamples. While we did not try to accurately estimate their frequency, our impression is that counterexamples were produced for less that 1 in 1000 test problems. The size of the asymptotic suboptimality of the index policy was no more than 0.002% in any example. Of course one should not place too much emphasis on results which depend on the way test problems are generated. We may be missing a class of examples for which the degree of suboptimality is greater. A better understanding might lead to more dramatic counterexamples, but the reasoning that led to the counterexample in section 4 does not seem to help. Nonetheless, the evidence so far is that counterexamples to the conjecture are rare and that the degree of suboptimality is very small. It appears that in most cases the index policy is a very good heuristic.

Acknowledgements

We wish to thank Dr Alan Weiss, of AT&T Bell Laboratories, who explained to us his interesting work on large deviation theory and Markov jump processes, and Prof Keith Glover, of Cambridge University Engineering Department's Control Group, who produced the numbers for the matrices (q^1_{ji}) and (q^2_{ji}) in the counterexample of section 4. These provided a more attractive counterexample than those found by randomly generating test problems.

Our research has been sponsored by National Science Foundation grant #ECS-8712798. The second author acknoowledges additional support from the FAW (Forschungs Institut für Anwendungsorientierte Wissensverarbeitung), Ulm, West Germany, for support during the summer of 1988.

References

Freidlin, M.I. and Ventsel, A.D. (1984) *Random Perturbations of Dynamical Systems.* Springer-Verlag, New York.

Gittins, J.C. and Jones, D.M. (1974) A dynamic allocation index for the sequential design of experiments. In Progress in Statistics (ed. J. Gani), 241-266, North Holland, Amsterdam.

Weiss, A. (1986) A new technique for analyzing large traffic systems, *Adv. Appl. Prob.* 18, 506-532.

Weiss, A. and Mitra, D. (1988) A transient analysis of a data network with a processor-sharing switch, *Bell Laboratories Technical Report.*

Whittle, P. (1988) Restless bandits: activity allocation in a changing world. in *A Celebration of Applied Probability* (ed. J. Gani), *J. Appl. Prob. Special Volume* 25A, 287-298.

Industrial and Systems Engineering
Report Series No. J-88-3
March, 1988
Revised October, 1989

APPROXIMATION RESULTS IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

*This research was supported in part by NSF grant ECS-8712798.

APPROXIMATION RESULTS IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

Abstract

We consider scheduling a batch of jobs with stochastic processing times on parallel machines. We derive various new formulae for the expected flowtime and weighted flowtime under general scheduling rules. Smith's Rule, which orders job starts by decreasing ratio of weight to expected processing time provides a natural heuristic for this problem. We obtain a bound on the worst case difference between the expected weighted flowtime under Smith's Rule and under an Optimal policy. For a wide class of processing time distributions this bound is of order O(1), and does not increase with the number of jobs.

*This research was supported in part by NSF grant ECS-8712798.

Approximation Results in Parallel Machines Stochastic Scheduling

Gideon Weiss
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Introduction:

One of the simplest yet most useful results in scheduling theory is that flowtime (sum of all the waiting and processing times of all the jobs) is minimized by the SPT (Shortest Processing Time First) rule. This result holds for a single processor (Smith, 1956) as well as for parallel processors (McNaughton, 1959). For the often more applicable weighted flowtime objective function (where the waiting and processing times are weighted by different cost per unit time for each job) the optimal sequence of jobs on a single processor is in decreasing order of weight to processing time ratios (Smith, 1956), the so called "Smith's Rule". Minimization of weighted flowtime on several parallel machines is, however, an NP-hard combinatorial optimization problem for any fixed number of machines greater than 1 (Garey and Johnson 1979, Lenstra, Rinnooy-Kan and Brucker 1977). Scheduling the starting times of the jobs according to Smith's rule provides a suboptimal heuristic for this problem, whose worst case performance is 1.2071 times the optimal value (Kawaguchi and Kyan 1986, Weiss et al. 1987).

In the present paper we examine a stochastic version of these problems, where we assume that the processing requirements of the jobs are not known in advance but are drawn from some known probability distributions. The single processor results generalize easily to stochastic processing times: SEPT (shortest expected processing time first) and Smith's Rule (decreasing order of weight to expected processing time ratio) respectively minimize the expected flowtime and the expected weighted flowtime.

Results for parallel machines are much more complex. SEPT remains optimal for flowtime in a wide range of problems, though the proofs are no longer elementary. SEPT is optimal when job processing times are exponentially distributed (Bruno, Downey and Frederickson 1981, Glazebrook 1979, Pinedo and Weiss 1979, Weiss and Pinedo 1980). It is also optimal when the job processing time distributions are all of them tails of a single IHR (increasing hazard rate) distribution (Weber 1982). Both these results are special cases of the remarkable result of Weber, Varaiya and Walrand (1986) that SEPT is optimal whenever the processing time distributions of all the jobs are stochastically comparable in pairs. That result seems the most general possible. It does not extend to weaker comparison conditions, and SEPT fails to be optimal in general (see Pinedo and Weiss 1987 for counter examples). Very little can be said on optimality of Smith's Rule for expected weighted flowtime on parallel machines: Kampke (1987) discusses some special cases.

If SEPT and Smith's Rule are no longer optimal on parallel machines, two questions arise: What is the optimal policy and how far are SEPT and Smith's Rule from it?

On the first question, I believe the search for the optimal policy in the general case is futile. First, the problem is NP-hard; it seems likely that one can find a parametric family of distributions for which even minimization of expected flowtime is NP-hard. But the difficulties go beyond the computational effort involved: if SEPT or Smith's Rule are not optimal then the optimal policy may be extremely complicated to describe and to implement - it may involve dynamic scheduling with inserted idle time, and may depend on the entire processing time distribution of each job rather than on a few parameters. Needless to say, the data to estimate these distributions in such detail will rarely be available in practice. Pinedo and Weiss (1987) discuss some very simple problems which have quite complicated optimal solutions; it is easy to imagine problems with optimal solutions of almost any degree of complexity.

How far SEPT and Smith's Rule are from optimal can be measured in two ways: in terms of the expected value of the objective function, or in terms of the differences between the policies (this difference can be expressed for example by counting the number of jobs which a realization of the optimal policy does not start according to Smith's Rule). In the present paper we give a complete answer in terms of the objective function. This answer is extremely favorable to Smith's Rule and SEPT.

Using the weight, the mean processing time, and one additional simply calculated parameter of the processing time distribution of each of the jobs, we calculate a bound on the difference between the expected objective function values. As long as the values of these three relevant parameters remain bounded, the bound on the expected difference does not grow with the number of jobs n. Thus under the assumption that the weights and the processing time distributions of all the jobs satisfy some uniform boundedness conditions we have that even though the expected (weighted) flowtime will as a rule increase as $O(n^2)$, the expected difference in objective value between SEPT (or Smith's Rule) and the optimal policy will remain bounded by a constant, independent of n.

The uniform boundedness is clearly essential here, without it the worst case performance ratio for deterministic jobs is ~1.2, so the difference is $O(n^2)$. However, an assumption of uniform boundedness on processing time distributions is in general much less restrictive than for the deterministic case. In many problems all that will be required is uniform bounds on the means and variances of the processing times: this still permits any mixture of actual processing time values to occur and seems a reasonable requirement.

The results of this paper are derived in sections 4-11. While deriving the bounds for the objective function we obtain some insights into the nature of parallel processing, and some useful formulas for expected flowtime and weighted flowtime. The results are summarized in section 2, and a plausibility argument is given in section 3.

With the question of how well SEPT or Smith's Rule will do in terms of the expected value of the objective function settled, it remains to ask how different are the actual policies from the optimal. The plausibility argument indicates that SEPT may not be optimal towards the end of the schedule. We conjecture that SEPT and Smith's Rule have a turnpike property (for a definition of turnpike optimality, see Shapiro, 1968) - asymptotically, for large n, most of the optimal decisions will be according to SEPT (or Smith's Rule). Such a turnpike result does indeed hold, and will be the subject of a forthcoming paper. A turnpike optimality result for a special case of a preemptive scheduling problem has been proven in Coffman, Hofri and Weiss (1988).

We conclude this paper with a discussion of some possible extensions to preemptive scheduling problems, the relation to Gittins index, and to various control problems in section 12.

2. Summary of Results

Jobs $1, \ldots, n$ require processing times $X_1, \ldots, X_n$, nonnegative, which are drawn independepently from some given probability distribution functions $F_1, \ldots, F_n$ but their actual values are not known in advance. Throughout this paper we assume that $X_1, \ldots, X_n$ are independent random variables, and that they are furthermore independent of when and how the jobs are performed, that is they are independent of the machines and of the schedule. We assume nothing special about the form of $F_1, \ldots, F_n$; We let $\mu_1, \ldots, \mu_n$ and $\sigma_1^2, \ldots, \sigma_n^2$ be their means and variances, and assume finite third moments. Machines $0, \ldots, M$ first become available to start the processing of these jobs at times $U_{00}, \ldots, U_{M0}$, with $U_{00} + \ldots + U_{M0} = 0$. Jobs are then processed by the machines in parallel, with no preemptions and no inserted idle times. Let $C_1, \ldots, C_n$ be the completion times of the jobs. Two objective functions are of interest: the flowtime $\sum_{j=1}^{n} C_j$ which is the sum of all the completion times (waiting plus processing times) of all the jobs, and, more generally, the weighted flowtime, $\sum_{j=1}^{n} W_j C_j$ where the jobs are weighted by individual costs per unit time, $W_1, \ldots, W_n$. Two policies suggest themselves for these objective functions: SEPT - start jobs in the order of shortest expected processing time first, and the so called "Smith's Rule" denoted by SR - start jobs in decreasing order of weight to expected processing time ratio. On a single processor, SEPT and SR minimize the expected flowtime and the expected weighted flowtime respectively. On $M+1$ parallel processors this is not generally the case.

In the present paper we prove that while SEPT and SR may not be optimal, they are very nearly optimal. In section 3 we give a plausibility argument for these results.

The results in section 4 are combinatorial in nature and apply to any processing times $X_1, \dots, X_n$, deterministic or stochastic.

(a) Decomposition of flowtime: Assume jobs are started in the order $1, \dots, n$, with processing times $X_1, \dots, X_n$, and are processed by machines $0, \dots, M$ with no preemptions or inserted idle time. In analogy with the definition of $U_{00}, \dots, U_{M0}$ we define for any $0 \leq m \leq n$: Let $U_{0m} \leq \dots \leq U_{Mm}$ be the ordered times at which the machines complete the service of the first m jobs to start. Let $D_{im} = U_{im} - U_{0m}$, $i = 1, \dots, M$ (where $m = 0, 1, \dots, n$) - these are the ordered remaining processing times of jobs on the M busy machines, when a machine becomes available for the start of the $m+1$'st job. Then the weighted flowtime can be decomposed as:

$$\sum_{j=1}^{n} W_j C_j = \frac{1}{M+1} \sum_{j=1}^{n} \left(\sum_{k=j}^{n} W_k \right) X_j + \frac{1}{M+1} \sum_{j=1}^{n} W_j \left(M X_j - \sum_{i=1}^{M} D_{ij-1} \right) \quad \text{§4(4.9)}$$

(b) The job processing time remainders D_{ij}, $i=1, \dots, M$, $j=0, 1, \dots, n$, can be calculated by a Markowian recursion: $D_{1j}, \dots, D_{Mj}$ are the ordered values of the M largest values minus the smallest value among $X_j, D_{1j-1}, \dots, D_{Mj-1}$. Let $S_j^2 = \frac{1}{M} \sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)} \left(\sum_{i=1}^{M} D_{ij} \right)^2$, for $0 \leq j \leq n$. Note that S_j^2 are the sample variances of $U_{0j}, \dots, U_{Mj}$, (or equivalently of $0, D_{1j}, \dots, D_{Mj}$); in particular, S_0^2, S_n^2 measure the "raggedness" of the beginning and the end of the multiprocessor schedule.

The D_{ij}'s satisfy the Key Formula:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1)\, S_0^2 - M(M+1)\, S_n^2 \qquad \text{§4(4.12)}$$

(c) Flowtime under SPT: If $U_{00} = \dots = U_{M0} = 0$, and jobs are started according to shortest processing time first, then:

$$\sum_{j=1}^{n} C_j = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\, X_j + \frac{1}{M+1} \sum_{i=1}^{M} i\, U_{in} \qquad \S4(4.18)$$

In sections 5-8 we derive various formulas for expected flowtime and expected weighted flowtime, under various assumptions. Proceeding from the special to the general we obtain the following results:

(d) If the job processing times $X_1, \dots, X_n$ are independent identically distributed with $E(X_j) = \mu$, $Var(X_j) = \sigma^2$, the expected flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{2(M+1)}\mu + \frac{n\,M}{2(M+1)}\mu\,(1 - \frac{\sigma^2}{\mu^2}) \qquad \S5(5.6)$$

$$- \frac{M}{2}\left(\frac{S_0^2}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right)$$

$$+ \frac{M}{2}\left(\frac{E(S_n^2)}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right)$$

where the first $O(n^2)$ term is the single machine flowtime speeded up (M+1) times, the second O(n) term incorporates steady state delays per job caused by parallel processing, the third O(1) term corresponds to non steady state initial conditions, and the last term, which because of the limiting behavior of $E(S_n^2)$ will under suitable conditions go to zero as $n \to \infty$ and hence be o(1), corresponds to non steady state final conditions. In section 5 we apply formula 5.6 to four examples of i.i.d. processing time distributions: deterministic, exponential, uniform, and DFR.

(e) If the job processing times $X_1, \dots, X_n$ are independent and all have the same mean, $E(X_j) = \mu$, but may have different variances $Var(X_j) = \sigma_j^2$, then for any work conserving (no inserted idle time) nonpreemptive schedule, the expected flowtime is:

$$E\left(\sum_{j=1}^{n} C_j\right) = \frac{n(n+1)}{2(M+1)}\,\mu + \frac{n\,M}{2(M+1)}\,\mu\,\left(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\,\mu^2}\right) - \frac{M}{2}\,\frac{S_0^2}{\mu} + \frac{M}{2}\,\frac{E(S_n^2)}{\mu}\,. \qquad \S6(6.1)$$

In this expression, the only part which depends on the schedule is $E(S_n^2)$. Apart from that term and the initial conditions given by S_0^2, the expression depends only on μ and on $\sum_{j=1}^{n} \sigma_j^2/n$. We show that in this case minimization of flowtime is equivalent to minimization of makespan in L_2.

In the following results (f-i) the job processing times $X_1,\ldots,X_n$ are independent but follow general distributions $F_1,\ldots,F_n$ with $E(X_j) = \mu_j$, $\mathrm{Var}(X_j) = \sigma_j^2$, $E(X_j^3) < \infty$.

(f) For any work conserving nonpreemptive schedule the expected weighted flowtime, for weights which are equal to the expected processing times $(W_j = \mu_j)$, is:

$$E\left(\sum_{j=1}^{n} \mu_j C_j\right) = \frac{1}{2(M+1)}\left(\sum_{j=1}^{n} \mu_j\right)^2 + \frac{1}{2}\sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)}\sum_{j=1}^{n} \sigma_j^2 - \frac{M}{2}S_0^2 + \frac{M}{2}E(S_n^2). \qquad \S7(7.1)$$

Again, the only term which depends on the schedule is $E(S_n^2)$.

(g) We can rewrite the general weighted flowtime in terms of the weights $\mu_1,\ldots,\mu_n$:

$$\sum_{j=1}^{n} W_j C_j = \frac{W_n}{\mu_n}\sum_{j=1}^{n} \mu_j C_j + \left(\frac{W_{n-1}}{\mu_{n-1}} - \frac{W_n}{\mu_n}\right)\sum_{j=1}^{n-1} \mu_j C_j + \cdots + \left(\frac{W_1}{\mu_1} - \frac{W_2}{\mu_2}\right)\sum_{j=1}^{1} \mu_j C_j. \qquad \S8(8.1)$$

Note that if jobs are indexed according to Smith's rule, that is $\frac{W_1}{\mu_1} \geq \frac{W_2}{\mu_2}$ $\ldots \geq \frac{W_n}{\mu_n}$, then all terms in (8.1) are nonnegative.

If jobs are started in the order $1,\ldots,n$ then each subset of the form $1,\ldots,k$ is scheduled with no inserted idle time, and by substituting (7.1) in (8.1) we get:

$$E\left(\sum_{j=1}^{n} W_jC_j\right) = \frac{1}{M+1} E\left(\sum_{j=1}^{n} W_jC_j \,\middle|\, \text{one machine}\right) + \frac{M}{2(M+1)} \sum_{j=1}^{n} W_j\mu_j\left(1-\frac{\sigma_j^2}{\mu_j^2}\right) + \frac{M}{2} \sum_{j=1}^{n} \frac{W_j}{\mu_j}\left[E(S_j^2) - E(S_{j-1}^2)\right] \qquad \S 8(8.2)$$

In sections 9-11 of the paper we derive bounds on the performance of the SR heuristics:

(h) Consider now several classes of strategies and let:

Π_0 - the strategy of starting jobs in the arbitrary given order $1,\ldots,n$ with no inserted idle time.

Π_1 - optimal list scheduling strategy.

Π_2 - optimal dynamic scheduling strategy.

Π_3 - optimal dynamic scheduling strategy with inserted idle time.

Π_4 - optimal dynamic scheduling strategy when the actual value of the processing time of a job becomes known when its processing starts.

Then,

$$E(\Sigma W_jC_j|\Pi_0) \geq E(\Sigma W_jC_j|\Pi_1) \geq E(\Sigma W_jC_j)|\Pi_2) \qquad \S9(9.5)$$
$$\geq E(\Sigma W_jC_j|\Pi_3) \geq E(\Sigma W_jC_j|\Pi_4)$$

(i) If we take the special weights $\mu_1,\ldots,\mu_n$, (7.1) holds for Π_0 as well as Π_4 and so:

$$0 \leq E(\sum_{j=1}^{n} \mu_jC_j|\Pi_0) - E(\sum_{j=1}^{n} \mu_jC_j|\Pi_4) \leq \frac{M}{2} E(S_n^2|\Pi_0). \qquad \S7(7.10)A$$

Applying (7.1) and (9.5) for the subset of jobs $1,\ldots,k$, where jobs $1,\ldots,k$ are the first k jobs in the schedule under Π_0, for any fixed k, $(1 \leq k \leq n)$ we get for an arbitrary strategy Π from any of the classes of strategies in d:

$$E(\sum_{j=1}^{k} \mu_jC_j|\Pi_0) - E(\sum_{j=1}^{k} \mu_jC_j|\Pi) \leq \frac{M}{2} E(S_k^2|\Pi_0). \qquad \S7(7.10)B$$

where S_k^2 as defined above is the sample variance of the times at which the machines finish jobs $1,\ldots, k$ under Π_0.

(j) Let

$$\bar{D}^2 = \max_{1\leq j\leq n} \max_{s>0} \int_s^\infty (x-s)^2 \, dF_j(x) \;/\; (1 - F_j(s)) \qquad \S10(10.1)$$

The value $\bar{D}^2$ is an upper bound on the second moment of the remaining processing time of any uncompleted job.

Then, if $S_0^2 = 0$,

$$E(S_k^2) \leq \frac{M}{M+1} \bar{D}^2 \qquad k = 1,\ldots,n \qquad \S10(10.2)$$

Combining (7.10), (10.2) and (8.1) we then have:

<u>Theorem</u>(§11,11.1): Let $\frac{W_1}{\mu_1} \geq \frac{W_2}{\mu_2} \geq \ldots \geq \frac{W_n}{\mu_n}$, then:

$$E(\Sigma W_jC_j|SR) - E(\Sigma W_jC_j|\Pi_4) \leq \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D}^2. \qquad \S11(11.1)$$

3. A Plausibility Argument

The main result of this paper, Theorem 11.1, is to show that the difference in the expected objective function when using SEPT or SR as a heuristic as against the optimal policy is, under suitable conditions, bounded by a constant. The suitable conditions are that there exists a uniform bound (as $n \to \infty$) on the second moments of residual processing time distributions, see (10.1) in the previous section for definition. This implies that the ratio

$$R = \frac{E(\text{objective} \mid \text{heuristic})}{E(\text{objective} \mid \text{optimal})}$$

converges to 1 as $n \to \infty$, whenever $E(\text{objective} \mid \text{optimal}) \to \infty$ as $n \to \infty$. As a rule one can expect that ΣC_j and $\Sigma W_j C_j$ increase like $\Omega(n^2)$, in which case the uniform bound on the difference implies $R = 1 + O(1/n^2)$. We now discuss the plausibility of such a result.

As a first step we show how to obtain $R = 1+O(\frac{1}{\sqrt{n}})$ behaviour.

Proposition 3.1: Let $E(X_1) \leq \ldots \leq E(X_n)$; Then

$$R = \frac{E(\text{Flowtime} \mid \text{SEPT})}{E(\text{Flowtime} \mid \text{OPT})} \leq 1+ \frac{2nM\, E(\max(X_1,\ldots,X_n))}{E(nX_1+\ldots+X_n) - nME(\max(X_1,\ldots,X_n))} \qquad (3.1)$$

Therefore, if $\sum_{j=1}^{n} (n-j+1)\, EX_j \sim \Omega(n^2)$, and $E(\max(X_1,\ldots, X_n)) \sim O(\sqrt{n})$ then $R = 1+ O(\frac{1}{\sqrt{n}})$.

Proof: Assume jobs are started in some arbitrary order $1,\ldots,n$ and C_m is the completion time of the m'th job on the M+1 parallel machines. Since all the jobs $1,\ldots,m-1$ start prior to job m, at time C_m there may be at most M jobs among $1,\ldots,m$ which are still running, and these remaining jobs run for at most $\max(X_1,\ldots,X_m)$. Thus the amount of work done on jobs $1,\dots,m$ up to time C_m is at least $\Sigma_{i=1}^{m}X_i$ - $M\max(X_1,\dots,X_m)$, and with $M+1$ processors to do the work C_m is at least $\frac{1}{M+1}\Sigma_{i=1}^{m}X_i - \frac{M}{M+1}\max(X_1,\dots,X_m)$. On the other hand, job m can start no later than $\frac{1}{M+1}\Sigma_{i=1}^{m}X_i - \frac{1}{M+1}X_m$ and so it has to complete at the latest by $\frac{1}{M+1}\Sigma_{i=1}^{m}X_i + \frac{M}{M+1}X_m$. We therefore have the celebrated result of Graham (1976) that:

$$\frac{1}{M+1}\sum_{j=1}^{m}X_j - \frac{M}{M+1}\max(X_1,\dots X_m) \le C_m \le \frac{1}{M+1}\sum_{j=1}^{m}X_j + \frac{M}{M+1}\max(X_1,\dots,X_m) \quad (3.2)$$

Summing over $m=1,\dots,n$ and recalling that $E(nX_1+\dots+X_n)$ is minimal if $E(X_1)\le\dots\le E(X_n)$, we have

$$E(\text{Flowtime}|\text{SEPT}) \le \frac{1}{M+1}E(nX_1+\dots X_n) + \frac{M}{M+1}n\,E(\max(X_1,\dots,X_n))$$

$$E(\text{Flowtime}|\text{OPT}\ ) \ge \frac{1}{M+1}E(nX_1+\dots X_n) - \frac{M}{M+1}n\,E(\max(X_1,\dots,X_n)),$$

and (3.1) follows. □

$E(nX_1+\dots+X_n)$ will be $\Omega(n^2)$ if jobs are i.i.d with finite mean $E(X_j)$ independent of n; if in addition we also have finite variance $V(X_j)$ independent of n, then $E(\max(X_1,\dots,X_n)) \le O(\sqrt{n})$. Downey (1989) has recently shown that for i.i.d jobs $E(\max(X_1,\dots,X_n)) \le O(n^{1/k})$ if $E(X_j^k) \le \infty$. One can therefore perhaps expect that $R = 1 + O(1/n)$. Similar behaviour can be argued for weighted flowtime and SR. It is therefore not surprising that SEPT and SR are asymptotically optimal under uniform boundedness conditions. However, we claim much faster asymptotics than the above proposition indicates. The plausibility argument in this section tries to explain that.

Consider scheduling deterministic jobs to minimize flowtime. The flowtime on a single machine, when jobs are scheduled in the order $1,\dots,n$ is $nX_1+(n-1)X_2+\dots+X_n$. In other words, during the processing of the 1st job, job 1, n jobs are waiting; during the processing of job

j. $(n-j+1)$ jobs are waiting, and during the processing of the last job only one job is waiting. Clearly we need to have $X_1 \le X_2 \ldots \le X_n$ for the optimal schedule.

For parallel machines, almost the same argument works, though it needs to be taken from last to first (see Figure 3.1)

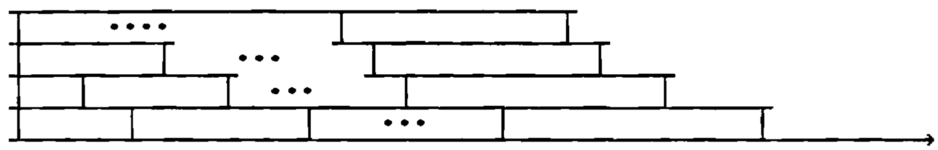


Figure 3.1: Flowtime on Parallel Machines.

Consider some schedule on M+1 parallel machines. Let J_1 be the set of all the jobs which are last on their machines; J_2 all the jobs which are before last, etc.: J_k are jobs which have k-1 jobs following them on the same machine. Assuming no idle times on any machine, the flowtime is: $\sum_{k=1}^{n} \sum_{j \in J_k} kX_j$. We therefore want to have the M+1 longest jobs in J_1, the next M+1 longest in J_2, etc; this yields the SPT schedule - start jobs according to shortest processing time first.

This argument does not work for stochastic processing times since the sets $J_1, J_2, \ldots$ cannot be assigned in advance.

For the sake of completeness we mention an alternative approach for the deterministic problem: Let $\tau_k < \tau_\ell$ be instants at which machines k and ℓ first become available; assume jobs are scheduled according to SPT; let n_k, n_ℓ be the number of jobs scheduled on machines k, ℓ respectively. Replace τ_k by $\tau_k+\Delta$ and τ_ℓ by $\tau_\ell-\Delta$; this will increase the flowtime by $\Delta(n_k-n_\ell)$. But $\Delta(n_k-n_\ell) \ge 0$ since under SPT, $n_k \ge n_\ell$. An inductive proof for optimality of SPT follows easily from this observation. This proof method can be adapted to stochastic processing times and is used by

Weber, Varaiya and Walrand (1986) to show that SEPT is optimal if processing times are comparable with respect to the stochastic order relation $\geq_{ST}$ (defined by $X \geq_{ST} Y$ if $P(X > x) \geq P(Y > x)$ for all x; equivalently, if $E(H(X)) \geq E(H(Y))$ for all nondecreasing H).

We return to the plausibility argument: The argument in favour of SEPT (as well as for Smith's Rule), is that at the beginning of the schedule there is a large number of jobs waiting and SEPT (or Smith's Rule) tend to reduce the number of jobs (their cost rate) fastest. This argument suffices to prove optimality for a single processor, and it applies to parallel processors as well. For parallel processors there exists, however, a counter argument: Towards the end of the schedule, as jobs are completed, there are no more new jobs to start and the processors fall idle one after the other; this means that processing at the end becomes inefficient and this of course has an effect on the objective function. Thus it seems that one ought to try to reduce these inefficiency periods. Minimization of these periods is hard (in the deterministic case, for two machines, it is equivalent to minimizing the makespan which is NP-hard), and it is not achieved by SEPT. If anything, it is asymptotically best to use LEPT to minimize makespan and the inefficiency periods (see Frenk and Rinnooy Kan, 1987). Nevertheless this inefficiency at the end is a boundary effect and is of marginal value - in particular it does not grow with the number of jobs n, unless the lengths of the jobs grow with n. Our conjecture is that this end effect is the only counter indication against optimality of SEPT (or Smith's Rule). Making this statement precise is the idea behind this paper.

4. Decomposition of Flowtime and of Weighted Flowtime

In this section we derive some useful formulas for decomposition of the flowtime and the weighted flowtime. Throughout this section we consider jobs $1,\ldots,n$ which require processing times $X_1,\ldots,X_n$ and are started in the order $1,\ldots,n$; jobs are processed without preemptions and with no inserted idle times. We let $C_1,\ldots,C_n$ denote their completion times.

The results in this section are for deterministic processing times and are of a combinatorial nature. We later regard them as sample path identities for the stochastic case. Much of the insight in these results follows by deriving them in easy steps: first for 1, then for 2, and finally for M+1 machines.

Case 1: Single Machine.

The completion time of job m is $C_m = X_1+\ldots+X_m$. Adding up we get for flowtime and weighted flowtime:

$$\sum_{j=1}^{n} C_j = nX_1+(n-1)X_2+\ldots+X_n \tag{4.1}$$

$$\sum_{j=1}^{n} W_j C_j = \sum_{j=1}^{n} \left(\sum_{k=j}^{n} W_k \right) X_j \tag{4.2}$$

Case 2: Two Machines:

We shall assume processing on the two machines starts at $U_0 \leq V_0$ with $U_0+V_0 = 0$, and let $D_0 = V_0-U_0$. Let $U_m \leq V_m$ be the times at which the two machines complete all the jobs $1,\ldots,m$, and let $D_m = V_m-U_m$. Clearly, $U_m+V_m = X_1 +\ldots+ X_m$. Job m+1 starts at U_m, and so:

$$C_{m+1} = U_m+ X_{m+1} = (U_m+V_m)/2 - D_m/2 + X_{m+1} = \tag{4.3}$$

$$= \frac{1}{2} (X_1+\ldots+X_{m+1}) + \frac{1}{2} (X_{m+1}-D_m).$$

Adding up we get:

$$\sum_{j=1}^{n} C_j = \frac{1}{2} (n X_1+(n-1) X_2 +\ldots+ X_n) \qquad (4.4)$$

$$+ \frac{1}{2} ((X_1-D_0) + (X_2-D_1) +\ldots+ (X_n-D_{n-1})).$$

and

$$\sum_{j=1}^{n} W_jC_j = \frac{1}{2} \sum_{j=1}^{n} (\sum_{k=j}^{n} W_k) X_j + \frac{1}{2} \sum_{j=1}^{n} W_j(X_j-D_{j-1}) \qquad (4.5)$$

We recognize the first part in these formulas as the flowtime (weighted flowtime) for a single machine working at twice the speed of (4.1), (4.2). The second part consists of individual job delays caused by parallel processing.

We now consider the sequence of D_m's. At time U_m when job m+1 starts on one machine, D_m is the remaining processing time of the last of the jobs 1,...,m, which is running on the other machine. It is easy to see that

$$D_{m+1} = |X_{m+1}-D_m|. \qquad (4.6)$$

If $X_1,\ldots, X_n$ are independent random variables, $\{D_m\}$ form a Markov chain.

The awkwardness of absolute value in (4.6) disappears when we square and sum over m.

$$\sum_{j=1}^{n} D_j^2 = \sum_{j=1}^{n} (X_j-D_{j-1})^2$$

from which we obtain the key formula

$$2 \sum_{j=1}^{n} X_jD_{j-1} = \sum_{j=1}^{n} X_j^2 + D_0^2 - D_n^2 \qquad (4.7)$$

<u>Case 3</u>: <u>M+1 machines</u>

Let $U_{00} \leq U_{10} \leq \ldots \leq U_{M0}$ be the times at which the M+1 machines first become available. We assume $\sum_{i=0}^{M} U_{i0} = 0$. For $m = 1,\ldots,n$, let

$U_{0m} \leq U_{1m} \leq \ldots, \leq U_{Mm}$ be the times at which the M+1 machines complete the processing of jobs $1,\ldots,m$; let $D_{im} = U_{im} - U_{0m}$, $i = 1,\ldots,M$, $m = 0,\ldots,n$.

Proposition 4.1: The flowtime on M+1 machines is

$$\sum_{j=1}^{n} C_j = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1) X_j + \frac{1}{M+1} \sum_{j=1}^{n} (MX_j - \sum_{i=1}^{M} D_{ij-1}) \tag{4.8}$$

and the weighted flowtime is:

$$\sum_{j=1}^{n} W_j C_j = \frac{1}{M+1} \sum_{j=1}^{n} (\sum_{k=j}^{n} W_k) X_j + \frac{1}{M+1} \sum_{j=1}^{n} W_j (MX_j - \sum_{i=1}^{M} D_{ij-1}) \tag{4.9}$$

Proof:

Job m+1 starts at U_{0m} and completes at $U_{0m} + X_{m+1}$. We have

$$\sum_{j=1}^{m} X_j = \sum_{i=0}^{M} U_{im} = (M+1)U_{0m} + \sum_{i=1}^{M} D_{im}.$$

Hence

$$C_{m+1} = \frac{1}{M+1} \sum_{j=1}^{m+1} X_j + \frac{M}{M+1} X_{m+1} - \frac{1}{M+1} \sum_{i=1}^{M} D_{im} \tag{4.10}$$

and (4.8), (4.9) follow. □

The Markovian recursion for D_{im}, $i=1,\ldots,M$ is:

Proposition 4.2: The values $D_{1m+1} \leq \ldots \leq D_{Mm+1}$ consist of the ordered values of the M largest among $X_{m+1}, D_{1m}, \ldots, D_{Mm}$ minus the smallest among $X_{m+1}, D_{1m}, \ldots, D_{Mm}$.

Proof: Follows from the fact that the completion time of jobs $1,\ldots, m+1$ on the M+1 machines occur at $U_{0m} + X_{m+1}, U_{1m}, \ldots, U_{Mm}$. □

The job processing time remainders D_{im}, $i = 1,\ldots,M$, $m = 0,\ldots,n$ can also be counted in a different way: Consider all the pairs of machines,

$0 \leq k < \ell \leq M$; for a particular pair k,ℓ let $n'(k,\ell)$ be the number of jobs performed on these machines. Let $j' = 1,\dots,n'(k,\ell)$ be an index counting the jobs in their starting order, let $X_{j'}^{(k,\ell)}$ be the processing times, and let $D_{j'}^{(k,\ell)}$ $j' = 0,\dots,n'(k,\ell)$ be the remaining processing times all relative to the pair of machines (k,ℓ). We show:

Proposition 4.3: There exists a 1-1 correspondence between the pairs $(X_{j'}^{(k,\ell)}, D_{j'-1}^{(k,\ell)})$ and the pairs (X_j, D_{ij-1}) where $j' = 1,\dots,n'(k,\ell)$, $0 \leq k < \ell \leq M$, and $j = 1,\dots,n$, $i = 1,\dots,M$.

Proof: Consider (X_j, D_{ij-1}). Let k be the machine on which job j is processed, let ℓ be the machine which at the start of job j has the remaining processing time D_{ij-1}. Consider the pair of machines k,ℓ; job j is performed on one of them (in fact on k); let j' be its index relative to the pair of machines k,ℓ. Then $(X_{j'}^{(k,\ell)}, D_{j'-1}^{(k,\ell)})$ is the pair which corresponds to (X_j, D_{ij-1}). This correspondence is clearly one to one. □

In particular:

$$\sum_{j=1}^{n} \sum_{i=1}^{M} D_{ij-1} = \sum_{0 \leq k < \ell \leq M} \sum_{j'=1}^{n'(k,\ell)} D_{j'-1}^{(k,\ell)} \tag{4.11}$$

We now show:

Proposition 4.4 - Key Formula: Let $S_j^2 = \frac{1}{M} \sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)} \left(\sum_{i=1}^{M} D_{ij}\right)^2$ (this is the sample variance of $0, D_{1j}, \dots, D_{Mj}$ or equivalently $U_{0j}, \dots, U_{Mj}$), $j = 0, 1, \dots, n$. Then:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1) S_0^2 - M(M+1) S_n^2 \tag{4.12}$$

Proof: By proposition 4.3:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = 2 \sum\sum_{0 \le k < \ell \le M} \sum_{j'=1}^{n'(k,\ell)} X_{j'}^{(k,\ell)} D_{j'-1}^{(k,\ell)}. \tag{4.13}$$

Applying (4.7) to each pair of machines this equals

$$\sum\sum_{0 \le k < \ell \le M} \left\{ \sum_{j'=1}^{n'(k,\ell)} (X_{j'}^{(k,\ell)})^2 + (D_0^{(k,\ell)})^2 - (D_{n'(k,\ell)}^{(k,\ell)})^2 \right\} \tag{4.14}$$

The summation over all pairs (k,ℓ) includes each X_j^2, $j = 1,\dots, n$, exactly M times. Also, $D_0^{(k,\ell)}$ equals $U_{k'0}-U_{\ell'0}$ for some pair of indices (k',ℓ') so that going over $0 \le k < \ell \le M$, $\{k',\ell'\}$ goes over all the pairs; Similarly, $D_{n'(k,\ell)}^{(k,\ell)}$ equals $U_{k''n}-U_{\ell''n}$. Hence (4.14) equals

$$M \sum_{j=1}^{n} X_j^2 + \sum\sum_{0 \le k < \ell \le M} \{(U_{k0}-U_{\ell 0})^2 - (U_{kn}-U_{\ell n})^2\} \tag{4.15}$$

Expression (4.12) follows from the well known statistics formula:

$$\sum_{1 \le s < t \le L} (a_s - a_t)^2 = L \sum_{s=1}^{L} a_s^2 - \left(\sum_{s=1}^{L} a_s\right)^2 \tag{4.16}$$

An Aside on SPT:

The decompositions (4.4), (4.8) allow us to get explicit formulas for flowtime when jobs are scheduled according to SPT. We look first at two machines. We let $U_0 = V_0 = 0$, and for SPT we have $X_1 \le X_2 \le \dots \le X_n$. Then the jobs start on the two machines in alternating order, $D_1 \le D_3 \le \dots$, and $D_2 \le D_4 \le \dots$, and the successive periods of length D_j are consecutive and not overlapping, see Figure 4.1.

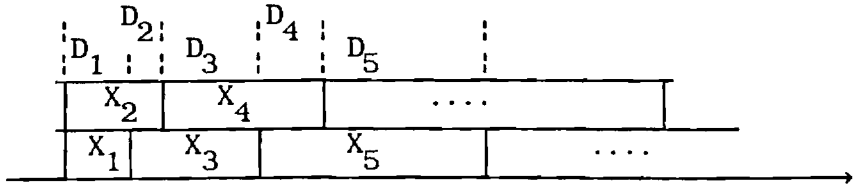


Figure 4.1: SPT on two machines

More precisely, with $D_0 = 0$, $D_1 = X_1$, and by induction $D_{j+1} = X_{j+1}-D_j$ (since if we assume $D_j = X_j-D_{j-1}$ then $D_j \leq X_j \leq X_{j+1}$); also $U_{j+1} = V_j$ $j = 0,\ldots,n-1$. Hence $\sum_{j=0}^{n-1} D_j = U_n$, and $\sum_{j=1}^{n} (X_j-D_{j-1}) = V_n$. V_n is however the makespan and so, by (4.4) we have, for 2 machines.

$$(\text{Flowtime}|\text{SPT on 2 machines}) = \tag{4.17}$$

$$= \frac{1}{2} (\text{Flowtime}|\text{SPT on 1 machine}) + \frac{1}{2} (\text{Makespan}|\text{SPT on 2 machines}).$$

On M+1 machines we have:

<u>Proposition 4.4:</u> Let $U_{00} = \ldots = U_{M0} = 0$ be the starting times of the M+1 machines, let jobs be scheduled by SPT so that $X_1 \leq \ldots \leq X_n$, and let $U_{0n} \leq \ldots \leq U_{Mn}$ be the finishing times for the M+1 machines. Then

$$\sum_{j=1}^{n} C_j = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\, X_j + \frac{1}{M+1} \sum_{i=1}^{M} i\, U_{in} \tag{4.18}$$

<u>Proof</u>: We use (4.8) and (4.11). To obtain $\sum_j \sum_i (X_j-D_{ij-1})$ as in (4.8), we look at all pairs of machines (k,ℓ). For a pair (k,ℓ) we calculate $\sum (X_{j'}^{(k,\ell)}-D_{j'-1}^{(k,\ell)})$. The $n'(k,\ell)$ jobs on these two machines are again scheduled by SPT, so the sum equals the makespan $V_{n'(k,\ell)}^{(k,\ell)}$ which is the finishing time of the later of these two machines. Summing over all pairs of machines, U_{Mn} gets counted M times, and in general U_{in} gets counted i times. The proposition follows. □

5. Flowtime for Random IID Jobs.

In this section we consider jobs whose processing times $X_1, \ldots, X_n$ are drawn independently from a common distribution F. For this case we obtain explicit formulas for the expected flowtime. We then apply these to get the expected flowtime for four specific examples of processing time distributions.

We let $\mu = E(X)$, $\sigma^2 = V(X)$ be the mean and variance of F, and assume $E(X^3) < \infty$; for simplicity we also assume in this section that F is non-arithmetic; we say that F is arithmetic (sometimes also called lattice) if it is concentrated on the set of points $0, \pm\lambda, \pm 2\lambda, \ldots,$ see Feller (1971, p136), otherwise F is nonarithmetic.

Imagine that there is an unlimited number of jobs which are scheduled on several parallel machines. The completion times of the jobs then form independent renewal processes. Consider an instant at which a job completion occurs on one machine, and examine the remaining processing times on the other machines. To be specific, let U_{i0}, $i = 0, \ldots, M$ be the initial starting times of the machines, and assume that a job is completed on machine k at time u. Observing the other machines at time u (conditional on $U_{10}, \ldots, U_{M0}$ and on u), the remaining processing times on the other M machines, $\{D_1, \ldots, D_M\}$, are independent forward recurrence times of the renewal processes, and for $u \to \infty$ they converge to M independent random variables identically distributed with the equilibrium distribution. Let D_∞ denote a random variable from the equilibrium distribution; it has probability density function $f_e(x) = \bar{F}(x)/\mu$ (where $\bar{F}(x) = 1-F(x)$), and 1st and 2nd moments $(\mu^2+\sigma^2)/2\mu$ and $E(x^3)/3\mu$ (Cox 1962).

Some care is needed in the preceding description: The independence hinges on the fact that we condition only on u. Independence still holds if we take the random instance U at which the completion of the jth job on machine k occurs, and condition on j. In contrast to this independence, if we condition on the event that the m'th job completion (counted on all the machines) occurs at time u (fixed or random), the remaining processing times on the other machines are no longer independent. Even conditioning on the event that the job completion at time u is of one of the first n jobs processed on all the machines destroys the independence; this point was raised by Van der Wal and Hordijk, (1987). Nevertheless, as $m \to \infty$, the joint distribution of the remaining processing times on the other processors, $D_{1m} \leq \dots \leq D_{Mm}$, converges to that of an ordered sample from the equilibrium distribution. To prove this convergence we need to examine the Markov chain of $D_{1m}, \dots, D_{Mm}$, $m = 0,1,\dots$, for iid X_m. Feller (1971, Chapter VI, Section II example f, p. 208) discusses the chain $D_{m+1} = |X_{m+1} - D_m|$, and Cox and Miller (1965, pp. 362-365) briefly discuss the multivariate chain; we were unable though to find a complete statement of the following theorem in the literature.

<u>Theorem 5.1</u>: For F nonarithmetic, $D_{1m}, \dots, D_{Mm}$ converge in distribution as $m \to \infty$ to $D_{1\infty}, \dots, D_{M\infty}$ with joint probability density function $f_e(\underline{x}) = \frac{M!}{\mu^M} \bar{F}(x_1) \dots \bar{F}(x_M)$, $x_1 \leq \dots \leq x_M$.

<u>Proof:</u> The proof is based on the approach in Feller, 1971, Chapter VIII, Section 7, pp. 270-274, which discusses ergodicity of Markov chains in discrete time with a general state space, and with a stationary transition kernel $K(\underline{x}, d\underline{y})$. Feller proves the following Theorem (Feller, op.cit, theorem 2): A strictly positive regular kernel K is ergodic if

and only if it possesses a strictly positive stationary probability distribution α. We use an adaptation of this theorem to show that the chain $D_{1m} \le \dots \le D_{Mm}$ is ergodic with convergence to the stated distribution $f_e(\underline{x})$. The proof requires the following steps:

(i) It is easily checked that $f_e(\underline{x})$ is the density of a strictly positive stationary distribution for the kernel K.

(ii) The kernel is regular (as defined in Feller, op. cit, definition 3) - this follows easily from the fact that our transition kernel operates like a convolution.

(iii) While for general nonarithmetic F the kernel is not strictly positive (as defined in Feller, op. cit, definition 1) it can be shown that it is asymptotically strictly positive in the sense:

Definition: The kernel $K(\underline{x}, d\underline{y})$ is asymptotically strictly positive in $\Omega \subseteq R^M$ if for every $\epsilon > 0$ there exists N so that for all $n > N$, $K^{(n)}(\underline{x}, I) > 0$ whenever the point $\underline{x}$ and the interval I satify: $\underline{x} \in \Omega \cap (0, 1/\epsilon)^M$, $I \subseteq \Omega \cap (0, 1/\epsilon)^M$, and $|P_i(I)| > \epsilon$ $i = 1, \dots, M$ ($P_i(I)$ is the projection of I on the i'th coordinate). This fact follows again from the convolution like nature of the kernel, utilizing a proof similar to Feller, Chapter V, section 4a, pp. 147-148.

(iv) The proof of Feller's theorem 2 can be modified to work when strictly positive is replaced by asymptotic strictly positive. The proof works in R^M as well as in R.

The proof is rather technical and given in the Appendix. □

For the next proposition we consider the limits of $E(\Sigma_{i=1}^{M} D_{in})$ and $(\Sigma_{i=1}^{M} D_{in}^2)$ as $n \to \infty$. Since we have that $(D_{1n}, \dots, D_{Mn})$ converge in distribution to $(D_{1\infty}, \dots, D_{n\infty})$, we may hope that moments of the D_{in}'s will converge to moments of $D_{i\infty}$. Existence of moments of the $D_{i\infty}$'s is clearcut: $E(D^k_{i\infty}) < \infty$ if and only if $E(D^k_\infty) < \infty$, which happens if and only if $E(X^{k+1}) < \infty$. However, existence of the moments and convergence in distribution does not in itself guarantee convergence of the moments. As is well known (Billingsley 1982), if $E(D^k_{i\infty}) < \infty$ and $D_{in} \xrightarrow{distr} D_{i\infty}$ as $n \longrightarrow \infty$ then a necessary and sufficient condition for $E(D^k_{in}) \longrightarrow E(D^k_{i\infty})$ is that D^k_{in} be uniformly integrable. A family of random variables Z_α, $\alpha \epsilon A$, with distributions F_α, is said to be uniformly integrable if

$$\lim_{a \to \infty} \sup_{\alpha \epsilon A} \int_a^\infty z \, dF_\alpha(z) = 0.$$

Unfortunately we were unable to ascertain under what conditions uniform integrability of D^k_{in} holds.

We note however two points: First, define

$$\bar{D}^{(k)} = \sup_{s>0} \frac{1}{1-F(s)} \int_s^\infty (x-s)^k \, dF(x) \tag{5.1}$$

(we define a similar quantity $\bar{\bar{D}}^2$ in section 10 (10.1)). Clearly, the condition that $\bar{D}^{(k)} < \infty$ is a sufficient condition for D^ℓ_{in} to be uniformly integrable for all $\ell \leq k$. Second, consider the renewal process with intervals distributed like X, let D_t be the forward recurrence time for this process at time t. Then the Key Renewal Theorem implies that $D_t \xrightarrow{distr} D_\infty$ as well as $E(D^k_t) \longrightarrow E(D^k_\infty)$ whenever $E(X^{k+1}) < \infty$. Thus the Key Renewal Theorem provides an indirect proof that if $E(X^{k+1}) < \infty$ then D^k_t are uniformly integrable. The similarity between the forward recurrence times D_t and our D_{in}'s leads us to believe that the condition $\bar{D}^{(k)} < \infty$ is overly stringent and $E(X^{k+1}) < \infty$ may in fact be sufficient.

We now turn to the value of $E(\sum_{j=1}^{n} \sum_{i=1}^{M} D_{ij-1})$ which appears in the flowtime expression (4.8). By Theorem 5.1, if we assume uniform integrability, $E(\sum_{i=1}^{M} D_{im}) \to M(\sigma^2+\mu^2)/2\mu$ as $m \to \infty$. In fact the convergence may be fast enough so that the deviations from the limit form a convergent series. We have:

<u>Proposition 5.2</u>: Under the assumption that D_{ij}^2 are uniformly integrable:

$$\lim_{n\to\infty} E \sum_{j=1}^{n} \sum_{i=1}^{M} \left(D_{ij-1} - \frac{\sigma^2+\mu^2}{2\mu}\right) =$$

$$= \frac{M(M+1)}{2} \frac{S_0^2}{\mu} - \frac{M^2 E(X^3)}{6\mu^2} + \frac{M(M-1)(\sigma^2+\mu^2)^2}{8\mu^3} \qquad (5.2)$$

<u>Proof</u>: We use the key formula (4.12), and take expectation on both sides:

$$E\, 2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = E\{ M \sum_{j=1}^{n} X_j^2 + M(M+1)S_0^2 - M(M+1)\, S_n^2 \}. \qquad (5.3)$$

On the left hand side we note that D_{ij-1} are functions of $X_1, \ldots, X_{j-1}$ only and so (whether j is fixed in advance or dependent on D_{ij-1}), X_j is distributed like F, independent of D_{ij-1}, $i=1,\ldots, M$, with $E X_j = \mu$. Hence:

$$2\mu\, E \sum_{j=1}^{n} \sum_{i=1}^{M} D_{ij-1} = n\, M(\sigma^2+\mu^2) + M(M+1)S_0^2 - M(M+1)E\, S_n^2. \qquad (5.4)$$

It remains to obtain $\lim_{n\to\infty} E\, S_n^2$, which by theorem 5.1 and the assumption of uniform integrablity is:

$$\lim_{n\to\infty} E\, S_n^2 = \frac{1}{M} E \sum_{i=1}^{M} D_{i\infty}^2 - \frac{1}{M(M+1)} E \left(\sum_{i=1}^{M} D_{i\infty}\right)^2 \qquad (5.5)$$

$$= \frac{M}{M+1} E(D_\infty^2) - \frac{M-1}{M+1} (E\, D_\infty)^2 = \frac{M}{M+1} \frac{E(X^3)}{3\mu} - \frac{M-1}{M+1} \left(\frac{\sigma^2+\mu^2}{2\mu}\right)^2,$$

where the second equality follows by changing the summation of the ordered sample to a summation over the unordered sample. The proposition follows. □

We combine the preceding results to obtain the main result of this section:

Theorem 5.3: The expected flowtime of n iid jobs on M+1 parallel machines is:

$$
\begin{aligned}
E\left(\sum_{j=1}^{n} C_j\right) &= \frac{n(n+1)}{2(M+1)}\mu + \frac{n\,M}{2(M+1)}\mu\left(1-\frac{\sigma^2}{\mu^2}\right) \\
&- \frac{M}{2}\left\{\frac{S_0^2}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right\} \\
&+ \frac{M}{2}\left\{\frac{E(S_n^2)}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right\}
\end{aligned}
\tag{5.6}
$$

Proof: This Theorem is a special case of Theorem 6.1, and the proof of Theorem 6.1 applies. However, if we assume uniform integrability of D_{ij}^2, j= 1,2,..., then we can use the following argument, which is more instructive. We rewrite the decomposition formula (4.8) and take expectations as follows:

$$
\begin{aligned}
E\left(\sum_{j=1}^{n} C_j\right) &= E\,\frac{1}{M+1}\sum_{j=1}^{n}(n-j+1)\,X_j \\
&+ E\,\frac{1}{M+1}\sum_{j=1}^{n}\left(M\,X_j - \sum_{i=1}^{M} D_{i\infty}\right) \\
&- E\,\frac{1}{M+1}\sum_{j=1}^{\infty}\left(\sum_{i=1}^{M} D_{i\,j-1} - \sum_{i=1}^{M} D_{i\infty}\right) \\
&+ E\,\frac{1}{M+1}\sum_{j=n+1}^{\infty}\left(\sum_{i=1}^{M} D_{i\,j-1} - \sum_{i=1}^{M} D_{i\infty}\right).
\end{aligned}
\tag{5.7}
$$

Substituting the expected value of X_j, of D_∞, and formula (5.2) we obtain (5.6). □

Note that the first term in (5.6) is the single machine flowtime for a machine with M+1 fold speedup. The second term contains an $\frac{M}{M+1}\mu\,(1 - \frac{\sigma^2}{\mu^2})$ delay per job that is the effect of parallel processing. If processing could start from stationary conditions this would be all. It is seen from (5.7) that the last two terms are the effect of starting in nonstationary conditions and of ending in nonstationary conditions. By (5.2) the last term will converges to 0 as $n \to \infty$ if we assume uniform integrability. The four terms are therefore $O(n^2)$, $O(n)$, $O(1)$ and $o(1)$ respectively. The whole expression is a function of μ, σ^2, S_0^2 and $E(S_n^2)$. Only the last of these depends on the form of the distribution.

Note: the assumption of uniform integrability which was used to prove proposition 5.2 is used in the above to verify that the last summand in (5.6) is of the order $o(1)$. It is not used anywhere else in the paper.

The following four examples further illustrate theorem 5.3.

Example 1 - Deterministic Jobs: $X_j = 1$, so $\sigma^2/\mu^2 = 0$. Assume jobs start with intervals of $\frac{1}{M+1}$ between them at $\frac{-M}{2(M+1)}, \frac{-M}{2(M+1)} + \frac{1}{M+1}, \frac{-M}{2(M+1)} + \frac{2}{M+1}, \ldots, \frac{-M}{2(M+1)} + \frac{M}{M+1}$; note that the sum of starting times on the M+1 machines is 0. Job completion times are at $\frac{M}{2(M+1)} + \frac{1}{M+1}, \frac{M}{2(M+1)} + \frac{2}{M+1}, \frac{M}{2(M+1)} + \frac{3}{M+1}, \ldots$. On a single M+1 fold speed machine completions would be at $\frac{1}{M+1}, \frac{2}{M+1}, \frac{3}{M+1}, \ldots$. Thus each job is delayed by $\frac{M}{2(M+1)}$. This delay per job is the maximal possible for any distribution F. Note that here $D_{im} = \frac{i}{M+1}$ and $S_0^2 = S_n^2$. See Figure 5.1.

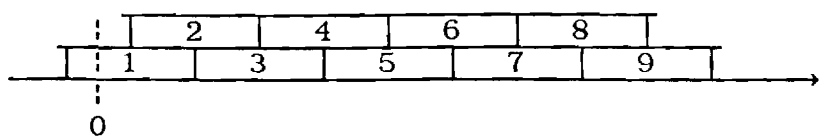


Figure 5.1: Deterministic Jobs on 2 Parallel Machines, n = 9.

The general formula is

$$E(\Sigma C_j) = \frac{n(n+M+1)}{2(M+1)}. \tag{5.8}$$

Example 2 - Exponential jobs: $X_j \sim \exp(1)$, $\sigma^2/\mu^2 = 1$. Assume jobs start at 0. While the machines are all busy successive job completions occur with ~ exp(M+1) intervals, exactly as on a M+1-fold speed single machine. There is no steady state delay per job - this is true whenever $\frac{\sigma^2}{\mu^2} = 1$. Here we chose $S_0^2 = 0$. D_{im}, i= 1,...M are an ordered sample from an exp(1) distribution, and $D_{im} \sim D_{i\infty}$. See Figure 5.2.

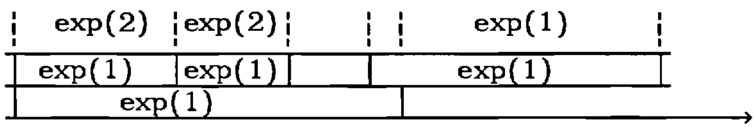


Figure 5.2: Exponential Jobs on 2 Parallel Machines, n = 4

The exact formula is:

$$E(\Sigma C_j) = \frac{n(n+1)}{2(M+1)} + \frac{M}{2}. \tag{5.9}$$

Example 3 - Uniform jobs: $X \sim U(0,2)$, $E(X) = 1$, $\sigma^2/\mu^2 = \frac{1}{3}$. For this distribution, D_∞ has p.d.f. $f_e(x) = 1 - \frac{1}{2}x$, $0 \leq x \leq 2$, and what's more, starting all the machines at time 0, the first job completion at a time > 0 already has D_{im} distributed like $D_{i\infty}$; see Feller, op. cit, problem 22, p. 217. The delay per job is $\frac{1}{3}\frac{M}{M+1}$. The formula for the flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n + \frac{2}{3}M + 1)}{2(M+1)} + \frac{M(M+2)}{9(M+1)} \tag{5.10}$$

Example 4 - DHR jobs: If X_j have a DHR (decreasing hazard rate, $h(x) = f(x)/\bar{F}(x)$ is decreasing), then $\sigma^2/\mu^2 > 1$. In that case the delay per job is negative. That is to say, the expected flowtime on M+1 parallel machines is smaller than on a single M+1 fold faster machine. This agrees with the result of Weber (1982) that in preemptive scheduling of DHR jobs processor sharing is optimal.

In general, the delay per job in comparison to a single M+1 fold faster machine is $\frac{M}{M+1}(1 - \frac{\sigma^2}{\mu^2})$. In the above examples we looked at $\frac{\sigma^2}{\mu^2} = 0, \frac{1}{3}, 1, >1$. The first three examples are special in that D_{im} actually become equal to $D_{i\infty}$, and so explicit formulas are obtained.

6. Jobs with Equal Mean Processing Times

In this section we consider jobs whose processing times $X_1, \ldots, X_n$ are drawn independently from distributions $F_1, \ldots, F_n$, all of them possessing the same mean $E(X_j) = \mu$; the jobs may however have different variances, $V(X_j) = \sigma_j^2$. On a single machine, the expected flowtime is $\frac{n(n+1)}{2}\mu$ for every nonpreemptive work conserving schedule. The minimization of expected flowtime on M+1 parallel machines is hard - they do not fall under the category of stochastically comparable jobs (as in Weber, Varaiya and Walrand 1986) and of course the SEPT rule is meaningless for such jobs, since every policy is SEPT. Pinedo and Weiss (1987) show that in some cases LVF (Largest Variance First) is optimal, but not in general.

We start by deriving a generalization of formula (5.6) for the expected flowtime. From this formula it becomes evident that the difference in expected flowtime between any two schedules lies in the "end effect" as conjectured in section 3, and this difference may be bounded by an O(1) quantity. We then discuss a connection with the minimization of expected makespan, which emerges from the formulas.

Theorem 6.1: The expected flowtime for n independent jobs with mean processing time μ and variances σ_j^2, $j = 1,\ldots,n$ scheduled on M+1 machines in the starting order $1,\ldots, n$ is given by:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{2(M+1)}\mu + \frac{nM}{2(M+1)}\mu(1 - \frac{\sum_{j=1}^{n}\sigma_j^2}{n\mu^2})$$

$$- \frac{M}{2}\frac{S_0^2}{\mu} + \frac{M}{2}\frac{E(S_n^2)}{\mu}. \tag{6.1}$$

Proof: We use the key formula (4.12), and take expectations on both sides, as in (5.3):

$$E\ 2\sum_{j=1}^{n}\sum_{i=1}^{M} X_j D_{ij-1} = E\{M\sum_{j=1}^{n} X_j^2 + M(M+1)S_0^2 - M(M+1)S_n^2\}. \tag{6.2}$$

Conditional on j, X_j is independent of D_{ij-1} and for all j, $E(X_j) = \mu$. Hence, (with $EX_j^2 = \mu^2 + \sigma_j^2$):

$$E\sum_{j=1}^{n}\sum_{i=1}^{M} D_{ij-1} = \frac{M\sum_{j=1}^{n}(\mu^2+\sigma_j^2)}{2\mu} + \frac{M(M+1)}{2\mu}S_0^2 - \frac{M(M+1)}{2\mu}E(S_n^2) \tag{6.3}$$

Taking expectations on the decomposition formula (4.8) we have:

$$E(\sum_{j=1}^{n} C_j) = E\{\frac{1}{M+1}\sum_{j=1}^{n}(n-j+1)\ X_j + \frac{M}{M+1}\sum_{j=1}^{n} X_j \tag{6.4}$$

$$- \frac{1}{M+1}\sum_{j=1}^{n}\sum_{i=1}^{M} D_{ij-1}\}.$$

Substituting $EX_j = \mu$ and (6.3) we obtain (6.1). □

As in (5.6), the first term in (6.1) is the expected flowtime for a single M+1 fold speed machine, the second term contains per job delays, with $\sum \sigma_j^2/n$ replacing σ^2 for the average delay, the third term is the effect of initial conditions and the last term is the end effect of the idle times at the end of the schedule. Only the last term depends on the form of the distributions $F_1,\ldots,F_n$, and only the last term depends on the schedule.

Consider any two scheduling strategies which are nonpreemptive and work conserving (i.e. do not allow idle time while unstarted jobs are available), Π_0 and Π, then:

$$E(\sum_{j=1}^{n} C_j|\Pi_0) - E(\sum_{j=1}^{n} C_j|\Pi) \leq \frac{M}{2\mu} E(S_n^2|\Pi_0). \qquad (6.5)$$

Expression (6.5) indicates that the difference in expected flowtime between any two schedules is bounded by a term which includes only the effect of the raggedness at the end of the schedule. We discuss bounds for this end effect in section 10.

Since the transition probabilities of the Markov chain of $D_{1m},\ldots,D_{Mm}$ are no longer stationary, the asymptotic behaviour of S_n is no longer as simple as in the i.i.d. case. Nevertheless, it is clear that as $n \to \infty$, the dependence of S_n on the first n_0 steps in the schedule tends to disappear.

From expression (6.1) it is seen that to minimize the expected flowtime is equivalent to minimizing $E(S_n^2)$.

Recall the definition of S_j^2 (proposition 4.4); S_j^2 is the sample variance of $0,D_{1j},\ldots,D_{Mj}$ or equivalently the sample variance of $U_{0j},U_{1j},\ldots,U_{Mj}$ and can written as:

$$S_j^2 = \frac{1}{M} \sum_{i=0}^{M} U_{ij}^2 - \frac{1}{M(M+1)} (\sum_{i=0}^{M} U_{ij})^2 \qquad (6.6)$$

We can now express the expected flowtime in terms of the machine completion times $U_{0n} \leq U_{1n} \leq \ldots \leq U_{Mn}$:

Proposition 6.2: The expected flowtime can be written as:

$$E(\sum_{j=1}^{n} C_j) = \frac{1}{2\mu} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{2\mu} \sum_{i=0}^{M} U_{i0}^2 + \frac{n}{2}\mu(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\mu^2}) \qquad (6.7)$$

<u>Proof:</u> Since we assume $\sum_{i=0}^{M} U_{i0} = 0$, we have $S_0^2 = \frac{1}{M} \sum_{i=0}^{M} U_{i0}^2$. To obtain an alternative expression for $E(S_n^2)$ we note that

$$\sum_{i=0}^{M} U_{in} = \sum_{j=1}^{n} X_j \tag{6.8}$$

so that

$$E(S_n^2) = \frac{1}{M} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{M(M+1)} E(\sum_{i=0}^{M} U_{in})^2 =$$

$$= \frac{1}{M} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{M(M+1)} E(\sum_{j=1}^{n} X_j)^2 \tag{6.9}$$

$$= \frac{1}{M} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{M(M+1)} (n^2\mu^2 + \sum_{j=1}^{n} \sigma_j^2).$$

Substituting in (6.1) we obtain (6.7). □

For two machines, M= 1, we have $S_n^2 = \frac{1}{2} D_n^2$ where D_n is the remaining processing time between the last two completion times. The expected flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{4}\mu + \frac{n}{4}\mu(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\mu^2}) - \frac{1}{4}\frac{D_0^2}{\mu} + \frac{1}{4}\frac{E(D_n^2)}{\mu}$$

on the other hand the makespan is:

$$C_{max} = \frac{1}{2} \sum_{j=1}^{n} X_j + \frac{1}{2} D_n$$

and so:

$$E(C_{max}) = \frac{n}{2}\mu + \frac{1}{2} E(D_n). \tag{6.10}$$

Hence, minimization of expected flowtime is equivalent to minimizing $E(D_n^2)$ while minimization of expected makespan is equivalent to minimizing $E(D_n)$. For many special cases, the same policy minimizes both, e.g. all the examples in Pinedo and Weiss (1987). For M+1 machines, we obtain:

<u>Corollary 6.3:</u> Minimization of expected flowtime is equivalent to minimization of the expected squared L_2 norm of $U_{0n}, U_{1n}, \ldots, U_{Mn}$.

In other words, when all $E(X_j)=\mu$, the minimization of expected flowtime is equivalent to a version of a stochastic makespan minimization problem.

For the deterministic problem minimizing makespan on two machines is the same as minimizing D_{1n}, which is the same as minimizing $S_n^2 = \frac{1}{2} D_n^2$ or minimizing $U_{On}^2 + U_{1n}^2$. For more than two machines several functions of $U_{On}, U_{1n}, \ldots, U_{Mn}$, the machine completion times, can be taken as generalizations of the two machine makespan. The natural one is to take $C_{max} = \max(U_{On}, \ldots, U_{Mn}) = U_{Mn}$, which is the L_∞ norm of $U_{On}, \ldots, U_{Mn}$. However, the L_2 norm, $\{\sum_{i=0}^{M} U_{in}^2\}^{1/2}$ is also a sensible measure for machine utilization. In the stochastic case the L_∞ and L_2 norms are random variables and we can try and minimize their expectations; for L_2, as often happens, minimization of the expected squared norm is more tractable. Our result here is that for equal mean processing times it is equivalent to minimizing flowtime.

7. A Special Case of Weighted Flowtime.

In this section we consider jobs with general random independent processing times, $X_1 \ldots, X_n$ with distribution functions $F_1, \ldots, F_n$, means $\mu_1, \ldots, \mu_n$ and variances $\sigma_1^2, \ldots, \sigma_n^2$. The simplicity of formulas (5.6) and (6.1) does not carry over to this general case. However, if we consider weighted flowtime with weights which are equal to the mean processing times, a straightforward generalization of (5.6), (6.1) is possible:

<u>Theorem 7.1</u>: The expected weighted flowtime when C_j - the completion time of job j, is weighted by μ_j - the expected processing time of job j is, for every nonpreemptive work conserving schedule:

$$E\left(\sum_{j=1}^{n} \mu_j C_j\right) = \frac{1}{2(M+1)} \left(\sum_{j=1}^{n} \mu_j\right)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 \tag{7.1}$$

$$- \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2)$$

Proof: Assume first that jobs are started in the predetermined order $1,\ldots,n$. Taking expectations in the key formula (4.12) we have:

$$2E\left(\sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{i\,j-1}\right) = M\, E\left(\sum_{j=1}^{n} X_j^2\right) + M(M+1)\, S_0^2 - M(M+1)\, E(S_n^2) \tag{7.2}$$

Because the order is predetermined, and $D_{i\,j-1}$ depends on $X_1, \ldots X_{j-1}$ only, X_j is independent of $D_{i\,j-1}$, $i = 1, \ldots, M$. Therefore:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} \mu_j\, E\, D_{i\,j-1} = M \sum_{j=1}^{n} (\mu_j^2+\sigma_j^2) + M(M+1)\, S_0^2 - M(M+1)\, E(S_n^2) \tag{7.3}$$

Taking expectation in the decomposition formula (4.9)

$$E \sum_{j=1}^{n} \mu_j C_j = \frac{1}{M+1} E\left\{ \sum_{j=1}^{n} \left(\sum_{k=j}^{n} \mu_k\right) X_j\right\} \tag{7.4}$$

$$+ \frac{M}{M+1} \sum_{j=1}^{n} \mu_j E(X_j) - \frac{1}{M+1} \sum_{j=1}^{n} \sum_{i=1}^{M} \mu_j\, E(D_{i\,j-1})$$

The first expectation is equal to $\sum_{j=1}^{n} \sum_{k=j}^{n} \mu_k\, \mu_j = \frac{1}{2} \left(\sum_{j=1}^{n} \mu_j\right)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2$, and further substitution yields (7.1).

Consider now a more general scheduling rule, under which the jobs are started in the order $J(1), J(2), \ldots, J(n)$, some permutation of $1, \ldots, n$. Here $J(1), J(2), \ldots$ can be random, and the j'th job to start, $J(j)$ can depend on $J(1), \ldots J(j-1)$ and on the values of $X_1, \ldots, X_{j-1}$ on which information is already available when the j'th job start occurs.

The decomposition formula (4.9) now reads:

$$\sum_{j=1}^{n} \mu_{J(j)} C_{J(j)} = \frac{1}{M+1} \sum_{j=1}^{n} \left(\sum_{k=j}^{n} \mu_{J(k)}\right) X_{J(j)} \tag{7.5}$$

$$+ \frac{1}{M+1} \sum_{j=1}^{n} \mu_{J(j)} \left(M\, X_{J(j)} - \sum_{i=1}^{M} D_{i\,j-1}\right)$$

The expectation of the first term adds up to:

$$\frac{1}{2(M+1)}\left(\sum_{j=1}^{n}\mu_j\right)^2+\frac{1}{2(M+1)}\sum_{j=1}^{n}\mu_j^2.$$

Also, by conditioning,

$$\begin{aligned} E(X_{J(j)}D_{ij-1}) &= E[D_{ij-1}\,E\{X_{J(j)}\,|D_{ij-1}\}] \\ &= E[D_{ij-1}\,E\{\,E(X_{J(j)}\,|D_{ij-1},J(j))\ \ |D_{ij-1}\}] \qquad (7.6) \\ &= E[D_{ij-1}\,E\{\mu_{J(j)}\,|D_{ij-1}\}] = E(\mu_{J(j)}D_{ij-1}), \end{aligned}$$

where the main point is that

$$E(X_{J(j)}\,|J(j),\ D_{ij-1}) = E(X_{J(j)}\,|J(j)) = \mu_{J(j)}. \qquad (7.7)$$

The key formula (4.12) now reads:

$$2\sum_{j=1}^{n}\ \sum_{i=1}^{M}(X_{J(j)}D_{ij-1}) = M\sum_{j=1}^{n}X_{J(j)}^2 + M(M+1)S_o^2 - M(M+1)\ E(S_n^2),$$

and taking expectations we have, by (7.6),

$$2\sum_{j=1}^{n}\ \sum_{i=1}^{M}E(\mu_{J(j)}D_{ij-1}) = M\sum_{j=1}^{n}(\mu_j^2+\sigma_j^2) + M(M+1)S_o^2 - M(M+1)E(S_n^2). \qquad (7.8)$$

Taking expectations in (7.5) and substituting (7.8) we again obtain (7.1). □

Several corollaries are easily obtained here:

Using the processing times X_j rather than their means μ_j as weights we have:

<u>Corollary 7.2</u>: For the case when C_j, the completion time of job j is weighted by X_j,

$$E\left(\sum_{j=1}^{n}X_jC_j\right) = E\left(\sum_{j=1}^{n}\mu_jC_j\right) + \sum_{j=1}^{n}\sigma_j^2 \qquad (7.9)$$

<u>Proof</u>: Consider (7.5); clearly in the new expected objective function $E(\sum_{j=1}^{n}X^2{}_{J(j)})$ replaces $E(\sum_{j=1}^{n}\mu_{J(j)}X_{J(j)})$. at the same time,

$E(X_{J(j)}X_{J(k)}) = E(\mu_{J(j)}\mu_{J(k)})$ for $k > j$, and $E(X_{J(j)}D_{ij-1}) = E(\mu_{J(j)}D_{ij-1})$, so nothing else is changed. (7.9) follows. □

With μ_j as weight for C_j, $j = 1,\ldots,n$, the expected weighted flowtime on a single machine is independent of the schedule. For $M+1$ parallel machines we have in analogy with (6.5):

Proposition 7.3: For any two nonpreemptive work conserving strategies, Π_o, Π

$$E\left(\sum_{j=1}^{n} \mu_j C_j | \Pi_o\right) - E\left(\sum_{j=1}^{n} \mu_j C_j | \Pi\right) \le \frac{M}{2} E(S_n^2 | \Pi_o). \tag{7.10}$$

Proof: By (7.1). □

We discuss bounds for $E(S_n^2)$ in Section 10.

In analogy with proposition 6.2 we have:

Corollary 7.4: In terms of machine start times $U_{00} \le \ldots \le U_{M0}$ where $\Sigma_0^M U_{i0}=0$ and finish times $U_{0n} \le \ldots \le U_{Mn}$:

$$E\left(\sum_{j=1}^{n} \mu_j C_j\right) = \frac{1}{2} E\left(\sum_{i=0}^{M} U_{in}^2\right) - \frac{1}{2} \sum_{i=0}^{M} U_{i0}^2 + \frac{1}{2}\left(\sum_{j=1}^{n} \mu_j\right)^2 - \frac{1}{2} \sum_{j=1}^{n} \sigma_j^2 \tag{7.11}$$

Proof: As in the proof of proposition 6.2, $S_0^2 = \frac{1}{M} \sum_{i=0}^{M} U_{i0}^2$ while

$$E(S_n^2) = E\left(\frac{1}{M} \sum_{i=0}^{M} U_{in}^2 - \frac{1}{M(M+1)} \left(\sum_{j=1}^{n} X_j\right)^2\right) \tag{7.12}$$

$$= \frac{1}{M} E\left(\sum_{i=0}^{M} U_{in}^2\right) - \frac{1}{M(M+1)} \left(\sum_{j=1}^{n} \mu_j\right)^2 - \frac{1}{M(M+1)} \sum_{j=1}^{n} \sigma_j^2$$

substituting in (7.1) we get (7.11) □

From (7.11) we see that minimization of weighted flowtime with weights μ_j is equivalent to minimization of $\sum_{i=0}^{M} E(U_{in}^2)$, as in section 6. Formula (7.11) for deterministic jobs appears in Eastman, Even and Isaacs (1964).

8. General Weighted Flowtimes.

In this section we consider general independent jobs and general weights. We derive an expression for the expected weighted flowtime. This generalizes (7.1), (6.1) and (5.6). It does not however share the simplicity of the previous formulas.

We start with the following useful decomposition.

Proposition 8.1: For any numbering of the jobs $1,\ldots,n$ with fixed weights $W_1,\ldots,W_n$, and with random job completion times $C_1,\ldots,C_n$ we have, no matter what the strategy is:

$$\sum_{j=1}^{n} W_j C_j = \frac{W_n}{\mu_n} \sum_{j=1}^{n} \mu_j C_j + \sum_{k=1}^{n-1} \left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \sum_{j=1}^{k} \mu_j C_j \tag{8.1}$$

Proof: immediate. □

We will use (8.1) in particular when one or both of the following two situations obtains: First situation - the indexing of the jobs $1,\ldots,n$ is such that the weight to expected processing time ratio is decreasing $\frac{W_1}{\mu_1} \geq \ldots \geq \frac{W_n}{\mu_n}$ (in other words, $1,\ldots,n$ is Smith's Rule ordering); Second situation - the jobs are started in the order $1,\ldots,n$ (in other words $1,\ldots,n$ is used as priority order for starting the jobs). In the first situation we have that $\left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \geq 0$ in (8.1). In the second situation we have that each of the summations $\sum_{j=1}^{k} \mu_j C_j$ is the weighted flowtime (with weights μ_j) of a subset of jobs $1,\ldots,k$ which are scheduled consecutively with no inserted idle times. Both situations obtain only if jobs are numbered by and scheduled according to Smith's Rule. Under the second situation the results of section 7 hold for any subset $\{1,\ldots,k\}$ and we have:

<u>Theorem 8.2</u>: If jobs are started in the order 1,..., n, then

$$E(\sum_{j=1}^{n} W_j C_j) = \frac{1}{M+1} E_1(\sum_{j=1}^{n} W_j \tilde{C}_j) + \frac{M}{2(M+1)} \sum_{j=1}^{n} W_j \mu_j (1 - \frac{\sigma_j^2}{\mu_j^2}) \tag{8.2}$$

$$+ \frac{M}{2} \sum_{j=1}^{n} \frac{W_j}{\mu_j} [E(S_j^{\,2}) - E(S_{j-1}^{\,2})]$$

Where $\tilde{C}_j$ are job completion times on a single machine, and E_1 is the expectation for scheduling weighted flowtime on a single machine.

<u>Proof</u>: Take expectations on (8.1), and substitute (7.1) to get:

$$E(\sum_{j=1}^{n} W_j C_j) = \frac{W_n}{\mu_n} [\frac{1}{2(M+1)}(\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2}\sum_{j=1}^{n} \mu_j^{\,2} - \frac{M}{2(M+1)}\sum_{j=1}^{n} \sigma_j^2 - \frac{M}{2}S_0^2 + \frac{M}{2}E(S_n^2)]$$

$$+ \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}})[\frac{1}{2(M+1)}(\sum_{j=1}^{k} \mu_j)^2 + \frac{1}{2}\sum_{j=1}^{k} \mu_j^{\,2} - \frac{M}{2(M+1)}\sum_{j=1}^{k} \sigma_j^2 - \frac{M}{2}S_0^2 + \frac{M}{2}E(S_k^2)]$$

$$= \frac{W_n}{\mu_n} \frac{1}{M+1} E_1(\sum_{j=1}^{n} \mu_j \tilde{C}_j) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{1}{M+1} E_1(\sum_{j=1}^{k} \mu_j \tilde{C}_j)$$

$$+ \frac{W_n}{\mu_n} \frac{M}{2(M+1)} \sum_{j=1}^{n} (\mu_j^2 - \sigma_j^2) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2(M+1)} \sum_{j=1}^{k} (\mu_j^2 - \sigma_j^2)$$

$$- \frac{W_1}{\mu_1} \frac{M}{2} S_0^2 + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2} E(S_k^2) + \frac{W_n}{\mu_n} \frac{M}{2} E(S_n^2)$$

from which (8.2) follows. □

We note again the structure of the expected weighted flowtime on parallel machines: The first term is the expected value for a single, M+1 fold speed machine. The second is a delay per job created by the variance of the job's processing time. The third term is more awkward than is (5.6), (6.1) and (7.1), it is a function of the sample variances of the remaining running times at each job start or completion; in (5.6), (6.1) and (7.1) only S_0^2 and $E(S_n^2)$ appeared, now all the $E(S_j^2)$ are present.

9. Some General Classes of Strategies.

In sharp contrast to the deterministic case, for stochastic processing times there are several classes of strategies, which give different optimal solutions. We discuss these classes and their optimal solutions here. We start with definitions.

Π_{-1} - The optimal strategy in the class of strategies which preassign jobs to machines.

SR - Smith rule strategy.

Π_o - Some arbitrary list strategy.

Π_1 - The optimal list strategy; the class of list strategies includes all strategies which predetermine the order in which jobs get started, and jobs are then scheduled without idle times (work conserving).

Π_2 - The optimal dynamic work conserving strategy; optimal in the class of strategies which upon every job completion choose a job to start immediately, based on current state.

Π_3 - The optimal dynamic strategy with inserted idle time; a strategy in this class allows the insertion of idle time at each job completion, and allows the choice of the job to start to be delayed until the end of the idle time.

Π_4 - The optimal dynamic strategy when the actual value of each X_j is revealed when the processing of job j starts.

We briefly discuss Π_{-1} here. The following result appears in Rothkopf, 1966; for more recent related work see also Lehtonen, 1988.

Proposition 9.1: The strategy Π_{-1} is the strategy which minimizes the deterministic problem with μ_j replacing X_j. The expected objective value of the former equals the objective value of the latter.

Proof: Let $J_i(1), \ldots, J_i(n_i)$ be the jobs assigned to machine i in the order in which they start. Then:

$$E\left(\sum_{j=1}^{n} W_j C_j\right) = E \sum_{i=0}^{M} \sum_{j=1}^{n_i} W_{J_i(j)} \sum_{k=1}^{j} X_{J_i(k)} \tag{9.1}$$

$$= \sum_{i=0}^{M} \sum_{j=1}^{n_i} W_{J_i(j)} \sum_{k=1}^{j} \mu_{J_i(k)}$$

which is the objective value of the deterministic problem with processing times μ_j. □

Hence, the problem of finding Π_{-1} for general expected weighted flowtime is NP-hard (Note! This is a stochastic problem which is NP-hard!). For the expected (unweighted) flowtime objective function, Π_{-1} is obtained by using the SPT schedule of deterministic jobs with processing times $\mu_1, \ldots, \mu_n$ to get $J_i(j)$, $j=1, \ldots, n_i$, $i=0, \ldots, M$. We have directly from (4.18):

Proposition 9.2: Let $\mu_1 \leq \ldots \leq \mu_n$, then for $S_0^2 = 0$,

$$E\left(\sum_{j=1}^{n} C_j \mid \Pi_{-1}\right) = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\,\mu_j + \frac{1}{M+1} \sum_{i=0}^{M} i\, E(\tilde{U}_{in}) \tag{9.2}$$

where $\tilde{U}_{in}$ are the machine completion times $i = 0, \ldots, M$, ordered by: $E(\tilde{U}_{0n}) \leq \ldots \leq E(\tilde{U}_{Mn})$ (in contrast to our usual order of machine completion times $U_{0n} \leq \ldots \leq U_{Mn}$).

Proof: By theorem 9.1 we need to calculate the value for a deterministic problem, with processing times μ_j. (9.2) then follows from (4.18) □

For jobs with equal expected processing times we have, if $n = L(M+1) + K$, $0 \leq K < M+1$, and $S_0^2 = 0$:

$$E\left(\sum_{j=1}^{n} C_j \mid \Pi_{-1}\right) = \frac{1}{M+1}\,\frac{n(n+1)}{2}\,\mu + \frac{1}{M+1}\,\frac{M(M+1)}{2}\,L\,\mu + \frac{1}{M+1}\,\frac{K(2M-k+1)}{2}\,\mu$$

and if n is a multiple of M+1,

$$E(\sum_{j=1}^{n} C_j|\Pi_{-1}) = \frac{n(n+1)}{2(M+1)}\mu + \frac{n\ M}{2(M+1)}\mu \tag{9.3}$$

so, for jobs with equal expected processing times, using (6.1):

$$E(\sum_{j=1}^{n} C_j|\Pi_{-1}) - E(\sum_{j=1}^{n} C_j|\Pi_0) = \frac{n\ M}{2(M+1)} \frac{\sum_{j=1}^{n}\sigma_j^2}{n\ \mu} - \frac{M}{2}\frac{E(S_n^2|\Pi_0)}{\mu} \tag{9.4}$$

the difference is of order O(n) and is typical for strategies which preassign jobs to machines. The reason for this is that if jobs are preassigned then when one machine completes all its processing, other machines will typically still have $O(\sqrt{n})$ jobs to run.

<u>Proposition 9.3</u>: For minimization of expected flowtime, $E(\sum_{j=1}^{n} C_j|\Pi_{-1}) \geq E(\sum_{j=1}^{n} C_j|\text{SEPT})$.

<u>Proof</u>: Consider the following strategy: Apply Π_{-1} to the set of n jobs, and start jobs accordingly. Whenever a job completes, consider all the remaining unstarted jobs, apply Π_{-1} to the remaining jobs, and start the next job accordingly. This iterated procedure is no worse than Π_{-1} since it minimizes over a wider class of possible schedues. But, for the minimization of $E(\Sigma C_j)$ this iterated strategy is identical to SEPT.

□

In general, for weighted flowtime, Π_{-1} is NP-hard and comparison with SR or with any other list strategy is not clearcut. We have in general:

<u>Proposition 9.4</u>:

$$\left.\begin{matrix} E(\Sigma W_j C_j|\Pi_0) \\ E(\Sigma W_j C_j|\Pi_{SR}) \end{matrix}\right\} \geq E(\Sigma W_j C_j|\Pi_1) \left.\begin{matrix} E(\Sigma W_j C_j|\Pi_{-1}) \\ \end{matrix}\right\} \geq E(\Sigma W_j C_j|\Pi_2) \geq E(\Sigma W_j C_j|\Pi_3) \geq E(\Sigma W_j C_j|\Pi_4) \tag{9.5}$$

<u>Proof</u>: The proof of the inequality for Π_{-1} uses the same arguments as

the proof of proposition 9.3. For all the other inequalities, the strategy on the right hand side of the inequality minimizes over a wider class of strategies than the one on the left hand side. □

The following proposition is required later:

Proposition 9.5: The strategy Π_4 does not use inserted idle time.

Proof: Assume that under strategy Π_4 a machine becomes available at time t_0. Let $t_1,\ldots,t_M$ with $t_0 \leq t_1 \leq \ldots \leq t_M$ be the times at which the other M machines next become available, following t_0; by our assumption $t_1,\ldots,t_M$ are known at t_0. Assume that strategy Π_4 inserts idle time at t_0 so that the next job start following t_0 is at $t_0+\Delta$, with job j starting. Without loss of generality job j starts on the machine that became available at t_0; call that machine machine 0.

Assume first that $t_0+\Delta > t_M$. Then the state at time $t_0+\Delta$ is identical to the state at time t_M which is earlier by $\delta = (t_0+\Delta-t_M)$. Change the strategy Π_4 to Π_4' which is identical to Π_4 for $t<t_m$, and for $t \geq t_M$ it does what Π_4 does at $t+\delta$. This saves δ on the completion time of each job that starts after t_0.

Assume next that $t_{k-1}<t_0+\Delta \leq t_k$, for some k, $1 \leq k \leq M$. Recall that job j is the first job to be started by Π_4 following t_0 and so between t_{k-1} and $t_0+\Delta$ no jobs start or end, and machine 0 (as well as the machines freed at $t_1,\ldots,t_{k-1}$) is idle in that interval. Modify Π_4 to Π_4' as follows: Instead of starting job j on machine 0 at time $t_0+\Delta$, start it on machine 0 at time t_{k-1} and then make machine 0 idle from $t_{k-1}+X_j$ until $t_0+\Delta+X_j$. Otherwise let Π_4' be the same as Π_4. This saves $\delta = (t_0+\Delta-t_{k-1})$ on the completion time of job j.

Hence in either case, if Π_4 uses inserted idle time we find Π_4' which improves on Π_4. The proposition follows. □

10. Bounds for Tails of Jobs

In the expressions for flowtime and weighted flowtime in Sections 5-8 the term $E(S_n^2)$, which is the expected value of the sample variance of the remaining processing times after U_{0n}, appears. We now discuss bounds on this term.

Theorem 10.1: Let $S_0^2 = 0$, and $X_1, \dots, X_n$ have distributions $F_1, \dots, F_n$. Let:

$$\bar{D}^2 = \max_{1 \leq j \leq n} \sup_{s > 0} \frac{1}{1-F_j(s)} \int_s^{\infty} (x-s)^2 \, dF_j(x) \tag{10.1}$$

Then, for $k = 1, \dots, n$, under any nonpreemptive work conserving strategy Π:

$$E(S_k^2 \mid \Pi) \leq \frac{M}{M+1} \bar{D}^2 \tag{10.2}$$

Proof: Consider any of the D_{ik}, $1 \leq i \leq M$, $1 \leq k \leq n$. D_{ik} is a remainder of a job that was started prior to U_{0k}, and has not completed its processing yet. If it is known which job D_{ik} consists of, and how long it has been processed prior to U_{0k}, then the distribution of D_{ik} is simply a tail of the distribution of this job. Condition on it being job j, and on job j having started at $U_{0k}-s$:

$$E(D_{ik}^2 \mid j, s) = E(D_{ik}^2 \mid D_{ik} = X_j - s,\ X_j > s) = \int_s^{\infty} (x-s)^2 \frac{dF_j(x)}{1-F_j(s)} \tag{10.3}$$

Hence,

$$E(D_{ik}^2) \leq \max_{1 \leq j \leq n} \sup_{s > 0} \int_s^{\infty} (x-s)^2 dF_j(x)/(1-F_j(s)) = \bar{D}^2. \tag{10.4}$$

Now,

$$S_k^2 = \frac{1}{M} \sum_{i=1}^{M} D_{ik}^2 - \frac{1}{M} \frac{1}{M+1} \Big(\sum_{i=1}^{M} D_{ik} \Big)^2$$

$$\leq \frac{1}{M} \Big(1 - \frac{1}{M+1}\Big) \sum_{i=1}^{M} D_{ik}^2 = \frac{1}{M+1} \sum_{i=1}^{M} D_{ik}^2 \tag{10.5}$$

and (10.2) follows. □

The following example shows that $\bar{D}^2$ may not be finite:

<u>Example 10.2:</u> X_j are distributed like X, a random variable with a Weibull distribution, with shape parameter $0 < \alpha < 1$. Then $F_j = F$,

$$F(x) = 1 - \exp(-x^{\alpha}) \qquad x \geq 0. \tag{10.6}$$

In this case, $Y = X^{\alpha}$ is distributed like an $\exp(1)$ random variable. Hence:

$$E(X^k) = E(Y^{k/\alpha}) = \Gamma(\tfrac{k}{\alpha} - 1) < \infty \tag{10.7}$$

so that X has moments of any order.

However, using integration by parts:

$$\frac{1}{1-F(s)} \int_s^{\infty} (x-s)\, dF(x) = \frac{\int_s^{\infty} e^{-x^{\alpha}}\, dx}{e^{-s^{\alpha}}}\, , \tag{10.8}$$

and using l'Hospital's rule,

$$\lim_{s \to \infty} \frac{\int_s^{\infty} e^{-x^{\alpha}}\, dx}{e^{-s^{\alpha}}} = \lim_{s \to \infty} \frac{e^{-s^{\alpha}}}{\alpha s^{\alpha-1} e^{-s^{\alpha}}} = \frac{1}{\alpha} \lim_{s \to \infty} s^{1-\alpha} = \infty \tag{10.9}$$

and we see that for this distribution, $D^{(1)}$ as defined in (5.1) is infinite. A similar calculation shows that $D^{(k)}$ is infinite for all $k > 0$ when $0 < \alpha < 1$, and is finite for all $k > 0$ when $1 \leq \alpha$. □

More generally, for a distribution F with density f and hazard rate function h:

$$\lim_{s \to \infty} \frac{\int_s^{\infty} (x-s)dF(x)}{\bar{F}(s)} = \lim_{s \to \infty} \frac{\int_s^{\infty} \bar{F}(x)dx}{\bar{F}(s)} = \lim_{s \to \infty} \frac{\bar{F}(s)}{f(s)} = \lim_{s \to \infty} \frac{1}{h(s)} \tag{10.10}$$

and

$$\lim_{s\to\infty} \frac{\int_s^\infty (x-s)^2 dF(x)}{\bar{F}(s)} = \lim_{s\to\infty} \frac{\int_s^\infty 2(x-s)\bar{F}(x)dx}{\bar{F}(s)} = \lim_{s\to\infty} \frac{\int_s^\infty 2\bar{F}(x)dx}{f(s)}$$

$$= \lim_{s\to\infty} \frac{2\bar{F}(s)}{f'(s)} = \lim_{s\to\infty} \frac{1}{h(s)} \frac{2}{\frac{d}{ds}\log(f(s))} \qquad (10.11)$$

so the finiteness of $D^{(1)}$, $D^{(2)}$ depends on the behaviour of $h(s)$ and of $\frac{d}{ds}\log(f(s))$ as $s\to\infty$.

Even if $D^{(2)}$ is finite for each of the distributions $F_1,\dots,F_n$, it is possible that $\bar{D}^{(2)}$ for the distribution F_n increases with n in such a way that $\bar{D}^2 \longrightarrow \infty$ as $n \longrightarrow \infty$. In many cases however, we can find simple bounds for the value of $\bar{D}^2$. We list some of these cases.

Case 1: $A \leq X_j \leq B$, the support of the distributions $F_1,\dots,F_n$ is bounded by B. In this case, clearly $\bar{D}^2 \leq B^2$.

Case 2: The distribution of X_j, $j= 1,\dots,n$ is one of $F_1,\dots,F_L$ where L is fixed and does not grow with n, and $D^{(2)}$ is finite for each of $F_1,\dots,F_L$. This seems to be a plausible behaviour in practice. Note that if $0< F_i(x)< 1$ for all $0< x< \infty$, then as $n \longrightarrow \infty$ $\max(X_j) \longrightarrow \infty$ and $\min(X_j) \longrightarrow 0$, but $\bar{D}^2$ does not grow with n.

Case 3: The distributions $F_1,\dots,F_n$ satisfy, for every $s>0$:

$$E(X^2) \geq E((X-s)^2|X>s) \qquad (10.12)$$

In this case:

$$\bar{D}^2 \leq \max_{1\leq j\leq n} E(X_j^2) = \max_{1\leq j\leq n} (\mu_j^2+\sigma_j^2).$$

The formula (10.12) defines a class of distributions which can be called "New better than used in second moments" (in analogy to NBUE New better than used in expectation, which is defined by $E(X) \geq E(X-s|X>s)$ for all $s>0$).

Case 4: The distributions $F_1, \dots, F_n$ are NBU - New Better than Used, defined by:

$$P(X > x) \geq P(X - s > x \mid X > s) \quad \text{all } x>0, \text{ all } s>0 \qquad (10.13)$$

(10.13) implies (10.12) so Case 4 is a special case of 3, with the same bound.

Case 5: The hazard rate of each X_j is bounded from below.

$$h_j(t) = \frac{dF_j(t)}{1-F_j(t)} \geq \lambda_j. \qquad (10.14)$$

In this case $X_j - s \mid X_j > s$ is stochastically smaller than an $\exp(\lambda_j)$ random variable, and so:

$$\bar{D}^2 \leq \max_{1 \leq j \leq n} \frac{2}{\lambda_j^2}. \qquad (10.15)$$

11. Approximate Optimality of SR

We now prove the approximate optimality result:

Theorem 11.1: Let $\frac{W_1}{\mu_1} \geq \dots \geq \frac{W_n}{\mu_n}$, and let $\bar{D}^2$ be defined as in section 10. Then:

$$E\left(\sum_{j=1}^{n} W_j C_j \mid SR\right) - E\left(\sum_{j=1}^{n} W_j C_j \mid \Pi_4\right) \leq \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D}^2. \qquad (11.1)$$

Proof: By (8.1) we can rewrite the difference as:

$$\Delta = \frac{W_n}{\mu_n} \left\{E\left(\sum_{j=1}^{n} \mu_j C_j \mid SR\right) - E\left(\sum_{j=1}^{n} \mu_j C_j \mid \Pi_4\right)\right\} \qquad (11.2)$$

$$+ \sum_{k=1}^{n-1} \left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \left\{E\left(\sum_{j=1}^{k} \mu_j C_j \mid SR\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \Pi_4\right)\right\}$$

Under SR, for the problem of scheduling jobs $1, \dots, n$ with weights $W_1, \dots, W_n$, jobs are started in the order $1, \dots, n$. Hence, for every k, $E\left(\sum_{j=1}^{k} \mu_j C_j \mid SR\right)$ is the expected value of the nonpreemptive, no idle time list policy which starts the jobs in the order $1, \dots, k$.

The value of $E(\sum_{j=1}^{k} \mu_j C_j | \Pi_4)$ is more complicated. Here Π_4 schedules all the jobs $1,\ldots,n$ to minimize $E(\sum_{j=1}^{n} W_j C_j)$ where no idle time is used (by Proposition 9.5), and where X_j becomes known when job j is started. If we now look at $\sum_{j=1}^{k} \mu_j C_j$ for the schedule given by Π_4, we may have jobs $1,\ldots,k$ started in some order different from $1,\ldots,k$ (and random), and with some of the jobs j, $j > k$, inserted in between. Let $\Pi_4(k)$ denote the optimal strategy for jobs $1,\ldots,k$ with weights $\mu_1,\ldots,\mu_k$. Clearly:

$$E(\sum_{j=1}^{k} \mu_j C_j | \Pi_4) \geq E(\sum_{j=1}^{k} \mu_j C_j | \Pi_4(k)) \tag{11.3}$$

Hence:

$$\Delta \leq \frac{W_n}{\mu_n} \{ E(\sum_{j=1}^{n} \mu_j C_j | SR) - E(\sum_{j=1}^{n} \mu_j C_j | \Pi_4) \} \tag{11.4}$$

$$+ \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \{ E(\sum_{J=1}^{k} \mu_j C_j | SR) - E(\sum_{j=1}^{k} \mu_j C_j | \Pi_4(k)) \}$$

since all the policies in (11.4) are now nonpreemptive and work conserving, by (7.10)

$$\Delta \leq \frac{W_n}{\mu_n} \frac{M}{2} E(S_n^2 | SR) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2} E(S_k^2 | SR) \tag{11.5}$$

Bounding $E(S_k^2 | SR)$ by (10.2)

$$\Delta \leq \frac{W_n}{\mu_n} \cdot \frac{M^2}{2(M+1)} \bar{D}^2 + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M^2}{2(M+1)} \bar{D}^2$$

$$= \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D}$$

□

12. Discussion.

In this paper we discussed the performance of Smith's Rule for nonpreemptive scheduling of a batch of jobs on parallel machines and have shown that its performance is very close to optimal. In the following discussion we briefly highlight the significance of our results, we list some possible extensions, and we explore some of the connections with other work in stochastic optimization.

Worst case performance of stochastic optimization heuristics: Heuristics for deterministic combinatorial optimiization problems are assessed by considering worst case performance or average performance. The former is often much worse than the latter; however, for average performance one needs to assume a distribution on the population of all possible problems, which may not be acceptable. In stochastic optimization problems, a distribution on the population of possible problems is part of the model. Expected worst case performance is in fact an average over this distribution of problems. This may suggest that the use of heuristics for stochastic optimiization problems can be more successful than their use for deterministic problems. Our paper is a case in point for this.

Preemptive scheduling of a batch of jobs on parallel machines. Preemptive scheduling of a batch of jobs on a single machine, to minimize weighted flowtime, is optimized by using a Gittins index policy (Gittins 1979, 1982). On parallel machines this suggests to schedule at any moment the jobs with the highest Gittins index as a suboptimal heuristic, analogous to the use of Smith's Rule in the nonpreemptive case. It would be interesting to assess its performance.

Scheduling of a stream of arriving jobs. If jobs with various processing time distributions arrive at a single server in independent Poisson streams, then Smith's Rule and the Gittins index policy remain optimal (see Sevcik 1974, Klimov 1974, Harrison 1975, Meilijson and Weiss 1977, Gittins and Nash 1977). Using these rules for parallel servers provides suboptimal heuristics. Clearly, there is now a nonoptimal end effect at the end of each busy period; nevertheless it may be possible to bound the worst case behavior of these heuristics.

Extensions to control of queueing networks. In Weiss (1988) Gittins type priority rules for scheduling customers in a queueing network which is served by a single server (the server jumps between the nodes of the network and provides preemptive service) are derived. These may provide some heuristics for more conventional networks in which all the nodes are served simultaneously by several servers in parallel.

Restless Bandits. Whittle (1987) has recently considered some generalizations of Gittins' original Bandit process model. In scheduling terms these can be expressed as including several parallel servers as well as exogenous changes in waiting jobs. Whittle suggests a Gittins' type heuristic for these processes, and conjectures that under the appropriate asymptotic conditions these may converge to optimal. Our results in this paper provide a special case for which Whittle's conjecture holds.

Queueing network heuristics based on diffusion approximations. Recently Wein (1987) has derived some heuristics for the control of queueing networks by considering heavy traffic conditions and using diffusion approximations. Some parts of these heuristics appear to be priority type rules to schedule several types of customers. It is intriguing to try and find a possible connection between our current work, Whittle's conjecture, and Wein's results.

References:

1.Billingsley, P., (1982). Measure and Probabillity. Wiley, New York.

2.Bruno, J., P.J. Downey, G. Frederickson, (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan, J. Association Computing Machines 28, 100-113.

3.Coffman, E.G., Jr., M. Hofri, G. Weiss, (1987). Scheduling stochastic jobs with a two point distribution on two parallel machines. Probability Engineering Information Sciences 3, 89-116.

4.Cox, D.R., (1962). Renewal Theory, Methuen, London.

5.Cox, D.R., H.D. Miller, (1965). Stochastic Processes, Methuen, London.

6.Downey, P.J., (1989). Distribution free bounds on the expectation of the maximum with scheduling applications. Operations Research Letters to appear.

7.Eastman, W.L., S. Even, I.M. Isaacs, (1964). Bounds for the optimal scheduling of n jobs on m processors. Management Science 11, 268-279.

8.Feller, W., (1971). An Introduction to Probability Theory and Its Applications, Vol. II, 2nd Edition, Wiley, N.Y.

9.Frenk, J.B.G., A.H.G. Rinnooy Kan (1987). The asymptotic optimality of the LPT rule. Mathematics Operations Research 12, 241-254.

10.Garey, M.R., D.S. Johnson (1979) Computers and Intractability: a Guide to the Theory of NP Completeness. Freeman, San Francisco.

11.Gittins, J. (1979) Bandit processes and dynamic allocation indices. J. Royal Statistical Society B 14, 148-167.

12.Gittins, J. (1982) Forwards induction and dynamic allocation indices. In Deterministic And Stochastic Scheduling. (M.A.H Dempster et al, eds.) Reidel, Dordrecht, 125-156.

13.Gittins, J., P. Nash (1977) Scheduling, queues, and dynamic allocation indices. Proc. EMS, Prague, 1974. Czechoslovak Academy of Sciences, Prague, 191-202.

14.Glazebook, F. D. (1979) Scheduling tasks with exponential service times on parallel processors, J. Applied Probability 16, 685-689.

15.Graham, R.L., (1976) Bounds on the performance of scheduling algorithms. In Computer and Job Shop Scheduling Theory E.G. Coffman Jr. Editor, Wiley, New York, 165-227.

16.Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimaly. Operations Research 23, 270-282.

17.Kampke, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines, J. Applied Probability 24, 430-448.

18.Kawaguchi, T., Kyan, S. (1986) Worst case bound of an LRF schedule for the mean weighted flowtime problem, SIAM J Computing 15, 1119-1129.

19.Klimov, G.P. (1974) Time-sharing service systems, I. Theory Probability Applications 19, 532-551.

20.Lenstra, J.K., A.H.G. Rinnooy-Kan, P. Brucker (1977) Complexity of machine scheduling problems, Annals Discrete Mathematics 1, 343-362.

21.Lehtonen, T., (1988) Scheduling jobs with exponential processing times on parallel machines, J Applied Probability 25, 752-762.

22.McNaughton, R. (1959) Scheduling with deadlines and loss functions, Management Science 6, 1-12.

23.Meilijson, I., G. Weiss, (1977) Multiple feedback at a single server station. Stochastic Processes Applications 5, 195-205.

24.Pinedo, M., G. Weiss, (1979). Scheduling stochastic tasks on two parallel processors, Naval Research Logistics Quarterly, 26, 527-536.

25.Pinedo, M., G. Weiss (1987) The largest variance first policy in some stochastic scheduling problems, Operations Research 35, 884-891.

26.Rothkopf, M. (1966) Scheduling with random service times, Management Science 12, 707-713.

27.Sevcik K.C. (1974) Scheduling for minimum total loss using service time distributions. J. Association Computing Machines 21, 66-75.

28.Shapiro, J.F. (1968) Turnpike planning horizons for a Markovian decision model, Management Science 14, 292-300.

29.Smith, W.E. (1956) Various optimizers for single stage production, Naval Research Logistics Quarterly, 3, 59-66.

30.Van der Wal, J., A. Hordijk (1987) Personal communication, OberWolfach meeting on Operations Research, February.

31.Weber, R.R. (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time, J. Applied Probability 19, 167-182.

32.Weber, R.R., P. Varaiya, J. Walrand, (1986) Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime, J. Applied Probability 23, 841-847.

33.Wein, L.M. (1987) Asymptotically optimal scheduling of a two station multiclass queueing network. PhD Dissertation, Stanford University.

34.Weiss, G. (1988) Branching bandit processes. Probability Engineering Informational Sciences, 2, 269-278.

35.Weiss, G., C. C. Huang, C. L. Li, S. Liu, M. Pinedo, J. Song (1987) A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime. Georgia Tech ISyE technical report series J-87-12.

36.Weiss, G., M. Pinedo, (1980) Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions, J. Applied Probability 17, 187-202.

37.Whittle, P. (1988) Restless bandits: activity allocation in a changing world, in A Celebration of Applied Probability J. Gani editor, J. Applied Probability special volume 25A, 287-298.

APPENDIX: Proof of Theorem 5.1.

In this appendix we prove Theorem 5.1 about the convergence of the Markov chain $D_{1m},\dots,D_{Mm}$, for F nonarithmetic. The proof closely follows similar derivations in Feller (1971). We assume that the iid processing times distribution F is a nonarithmetic distribution. We let $B = \sup\{x|F(x) < 1\} \le \infty$, and we let $\Omega = \{y_1,\dots, y_M | 0 \le y_1 \le y_2 \le \dots \le y_M \le B\} \subset R^M$ be the sample space for $D_{1m},\dots, D_{Mm}$: we say I is an open interval in Ω if $I \subseteq \Omega$ is of the form $(\alpha_1,\beta_1)\times\dots\times(\alpha_m,\beta_m)$ and say $|I| < \epsilon$ if $|\beta_i-\alpha_i| < \epsilon$, $i= 1,\dots, M$, and $|I| > \epsilon$ if $|\beta_i-\alpha_i| > \epsilon$, $i=1,\dots,M$. Let $\gamma_m(d\underline{y})$, $m = 0,1,\dots$ denote the probability distribution of $D_{1m},\dots,D_{Mm}$, and let $K(\underline{x},d\underline{y})$ denote the transition kernel of the chain, so that (Feller, Chapter VIII, Section 7, p. 270.)

$$\gamma_{m+1}(d\underline{y}) = \int K(\underline{x},d\underline{y})\ \gamma_m(d\underline{x}), \tag{A.1}$$

describes the one step Markov transitions, and let $K^{(n)}(\underline{x},d\underline{y})$ be the n step transition kernel, from γ_m to γ_{m+n}.

For our process the one step transition is given by (here dx denotes an infinitesimal interval around x):

$$\gamma_{m+1}(dy_1,\ldots, dy_M) = \int_o^\infty \gamma_m(d(y_1+u),\ldots,d(y_M+u))\ dF(u)$$

$$+ \int_o^\infty \gamma_m(du,d(y_2+u),\ldots,d(y_M+u))\ dF(y_1+u) \qquad (A.2)$$

$$\vdots$$

$$+ \int_o^\infty \gamma_m(du,d(y_1+u),\ldots,d(y_{j-1}+u),d(y_{j+1}+u),\ldots,d(y_M+u))\ dF(y_j+u)$$

$$\vdots$$

$$+ \int_o^\infty \gamma_m(du,d(y_1+u),\ldots,d(y_{M-1}+u))\ dF(y_M+u).$$

for $y_1 \leq \ldots \leq y_M$.

We recall some of the definitions from Feller:

Definition A.1: (Feller, Chapter VI, section II, Definition 2, p. 207)

A measure α is called a stationary measure for the kernel K if $\alpha = \alpha_o = \alpha_1 = \ldots \alpha_m = \alpha_{m+1} = \ldots$ in the transition relation (A.1).

Definition A.2: (Feller, Chapter VIII, section 7, Definition 1, p. 271)

A measure α is strictly positive in Ω if $\alpha(I) > o$ for every open interval in Ω.

Definition A.3: (Feller, op cit, definition 3) The kernel K is ergodic if there exists a strictly positive probability distribution α on Ω such that $\gamma_n \to \alpha$ independently of the initial probability γ_o.

Dual to (A.1) we also look at the relation:

$$U_{n+1}(\underline{x}) = \int K(\underline{x},d\underline{y})\ U_n(\underline{y}) \qquad (A.3)$$

which for our process has the form:

$$U_{n+1}(x_1,\dots,x_M) = \int_0^{x_1} U_n(x_1-u,x_2-u,\dots,x_M-u)\ dF(u) \tag{A.4}$$

$$+ \int_{x_1}^{x_2} U_n(u-x_1,x_2-x_1,\dots,x_M-x_1)\ dF(u)$$

$$\vdots$$

$$+ \int_{x_j}^{x_{j+1}} U_n(x_2-x_1,\dots,x_j-x_1,u-x_1,x_{j+1}-x_1,\dots,x_M-x_1) dF(u)$$

$$\vdots$$

$$+ \int_{x_M}^{\infty} U_n(x_2-x_1,\dots,x_M-x_1,u-x_1)\ dF(u).$$

for $x_1 \leq \dots \leq x_M$.

Definition A.4: (Feller, op cit, definition 3) The kernel K is regular if whenever U_0 is uniformly continuous (so that for every $\epsilon > 0$ there exists $\delta > 0$ such that if $\|\underline{x}'-\underline{x}''\| < \delta$, $\underline{x}',\underline{x}'' \in \Omega$ then $|U_0(\underline{x}')-U_0(\underline{x}'')| < \epsilon$) the whole family of functions U_n, $n = 0,1,\dots$ defined by A.3 is equicontinuous (so that for every $\epsilon > 0$ there exists $\delta > 0$ such that if $\|\underline{x}'-\underline{x}''\| < \delta$, $\underline{x}',\underline{x}'' \in \Omega$ then $|U_n(\underline{x}')-U_n(\underline{x}'')| < \epsilon$).

Definition A.5: (Feller, op cit, definition 2) The kernel K is strictly positive if $K(\underline{x},I) > 0$ for every $\underline{x} \in \Omega$ and every open interval $I \subseteq \Omega$.

To these we add:

Definition A.6: The kernel K is asymptotically strictly positive if for every $\epsilon > 0$ there exists N such that for all $n > N$, $K^{(n)}(\underline{x},I) > 0$ for all $\underline{x} \in \Omega$ which satisfies $0 < x_i < 1/\epsilon$, and all open intervals I, $I \subseteq \Omega$ which satisfy $|I| > \epsilon$, and which are contained in $(0,1/\epsilon)^M$.

We want to prove that the chain $D_{1m}, \ldots, D_{Mm}$ is ergodic, with $\alpha(d\underline{x}) = \frac{M!}{\mu^M} \bar{F}(x_1) \ldots \bar{F}(x_M)\, d\underline{x}$ for $\underline{x} \in \Omega$ as the limiting distribution. The proof is in four steps.

- α is a stationary distribution for the kernel K.
- The kernel K is regular.
- The kernel K is asymptotically strictly positive.
- The above three points imply ergodicity

The last point is an extension of Feller, op cit, theorem 2, p. 272 in which asymptotically strictly positive replaces strictly positive.

Proposition A.1: The measure $\alpha(d\underline{y}) = \frac{M!}{\mu^M} \bar{F}(y_1) \ldots \bar{F}(y_M) dy_1 \ldots dy_M$ for $\underline{y} \in \Omega$ is a stationary measure for the chain $D_{1m}, \ldots, D_{Mm}$, $m = 0, 1, \ldots$.

Proof: Substitute in the right hand side of (A.2) to obtain, for $y_1 \leq \ldots \leq y_M$:

$$\left(\frac{M!}{\mu^M} dy_1 \ldots dy_M\right) \times \int_0^\infty \bar{F}(y_1+u) \ldots \bar{F}(y_M+u)\, dF(u)$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_2+u) \ldots \bar{F}(y_M+u)\, dF(y_1+u)$$

$$\vdots$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_1+u) \ldots \bar{F}(y_{j-1}+u)\, \bar{F}(y_{j+1}+u) \ldots \bar{F}(y_M+u)\, dF(y_j+u)$$

$$\vdots$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_1+u) \ldots \bar{F}(y_{M-1}+u)\, dF(y_M+u) =$$

$$= \frac{M!}{\mu^M} dy_1 \ldots dy_M \int_0^\infty - d(\bar{F}(u)\bar{F}(y_1+u) \ldots \bar{F}(y_M+u))$$

$$= \frac{M!}{\mu^M} dy_1 \ldots dy_M\, \bar{F}(y_1) \ldots \bar{F}(y_M). \qquad \square$$

Proposition A.2: The kernel K described by (A.2) is regular.

Proof: The proof for two machines (M = 1) follows Feller (op cit, example a, p. 272). Assume for given $\epsilon > 0$ there is $\delta < 0$ such that if $\|x'-x''\| < \delta$ then $|U_m(x')-U_m(x'')| < \epsilon$. We now show that this implies (for the same ϵ): $|U_{m+1}(x')-U_{m+1}(x'')| < \epsilon$. Assume $x' < x''$.

$$|U_{m+1}(x')-U_{m+1}(x'')| \le \int_o^{x'} |U_m(x'-y) - U_m(x''-y)| \, dF(y)$$

$$+ \int_{x'}^{x''} |U_m(y-x') - U_m(x''-y)| \, dF(y)$$

$$+ \int_{x''}^{\infty} |U_m(y-x') - U_m(y-x'')| \, dF(y)$$

Note that for $x' \le y \le x''$, $|(y-x') - (x''-y)| \le |y-x'| + |x''-y| = |x''-x'| < \delta$ and of course $|(x'-y) - (x''-y)| < \delta$, $|(y-x') - (y-x'')| < \delta$, so that the whole expression is $\le \int \epsilon \, dF(y) = \epsilon$.

This proof fails for M > 1, since the induction step involves $U_m(y-x_1', x_2'-x_1', \dots) - U_m(y-x_1'', x_2''-x_1'', \dots)$ and the distance in norm between the arguments may double.

However, the proposition still holds, by a different proof. For $\epsilon > 0$ let $\delta > 0$ satisfy: $\|\underline{x}'-\underline{x}''\| < \delta$ implies $|U_o(\underline{x}') - U_o(\underline{x}'')| < \epsilon$. Use now $\delta/2$, and take any $\underline{x}'$, $\underline{x}''$ such that $\|\underline{x}'-\underline{x}''\| < \delta/2$. Consider $U_n(\underline{x}')$, $U_n(\underline{x}'')$. They equal the values of U_o evaluated at the points reached from $\underline{x}'$, $\underline{x}''$ after n transitions. Let $X_1, \dots, X_n$ drawn independently from F be the random variables which define the n step transition (the processing times). Condition on $X_1 = y_1, \dots, X_n = y_n$; define $\tilde{\underline{x}}_o' = (0, x_1', \dots, x_n')$, $\tilde{\underline{x}}_o'' = (0, x_1'', \dots, x_n'')$ and let $\tilde{\underline{x}}_{i+1}'$, $\tilde{\underline{x}}_{i+1}''$ be defined inductively as the vectors obtained by adding y_{i+1} to the smallest components of $\underset{\sim}{\tilde{x}}_i'$, $\tilde{\underline{x}}_i''$ and reordering the M+1 components. Then $\|\tilde{\underline{x}}_n' - \tilde{\underline{x}}_n''\| < \delta/2$. Obtain $\underline{x}_n'$, $\underline{x}_n''$ by subtracting the first component from the last M components of $\tilde{\underline{x}}_n'$, $\tilde{\underline{x}}_n''$. Then $\|\underline{x}_n'-\underline{x}_n''\| < \delta$. Hence, $|U_0(\underline{x}_n') - U_0(\underline{x}_n'')| < \epsilon$. But $U_0(\underline{x}_n') = U_n(\underline{x}'|\underline{y})$, $U_0(\underline{x}_n'') = U_n(\underline{x}''|\underline{y})$. Taking expectations, we get $|U_n(\underline{x}') - U_n(\underline{x}'')| < \epsilon$. □

<u>Proposition A.3</u>: The kernel K is asymptotically strictly positive.

<u>Proof</u>: We give first the proof for 2 machines, M = 1. We proceed similarly to Feller (1971, Chapter V, Section 4a, pp. 147-148). Recall that y is a point of increase of a distribution F if $F\{I\} > 0$ for every open interval I containing y; a point of increase of K(x,dy) and of the n stage transition $K^{(n)}(x,dy)$ is defined similarly. Let $\epsilon > 0$ be given. We start by choosing $b \geq \min(B, 1/\epsilon)$ such that b is a point of increase of F (for B finite take b = B). We take a fixed arbitrary x, $x < \min(b, 1/\epsilon) \leq b$. Clearly x is (the only) point of increase of $K^{(0)}(x,dy)$ and b-x is a point of increase of $K^{(1)}(x,dy)$. Also, if $y \leq b$ is a point of increase of $K^{(n)}(x,dy)$, then y is also a point of increase of $K^{(n+2)}(x,dy)$. Hence, x (b-x) is a point of increase of $K^{(n)}(x,dy)$ for all n which are even (odd).

Since F is nonarithmetic we can find a point of increase of F, say a, such that $0 < kb-\ell a = h < \epsilon$, with $k+\ell$ even. By considering $x + X_{i_1} + \ldots + X_{i_\ell} - X_{j_1} - \ldots - X_{j_k}$ we see that x-h is a point of increase of $K^{(k+\ell)}(x,dy)$, and of $K^{(n)}(x,dy)$ for n even, $n \geq (k+\ell)$. Similarly, considering $x+X_{j_1} + \ldots + X_{j_k} - X_{i_1} - \ldots - X_{i_\ell}$, we see that x+h is a point of increase of $K^{(k+\ell)}(x,dy)$, and of $K^{(n)}(x,dy)$ for n even, $n \geq (k+\ell)$. In the same way, (b-x) - h and (b-x) + h are points of increase of $K^{(n)}(x,dy)$ for all n odd, $n \geq k+\ell+1$. Next one sees that for n even,

$n \geq 2(k+\ell)$, all of $0 < x-2h,\ x-h,\ x,\ x+h,\ x+2h < b$ are points of increase of $K^{(n)}(x,dy)$, with similar statement for b-x. Take $N = \lceil \frac{b}{h} \rceil (k+\ell)$; then for n even, $n \geq N$, all points of the form $0 < x \pm mh < b$ are points of increase of $K^{(n)}(x,dy)$ while for n odd, $n \geq N$, all points of the form $0 < b-x \pm mh < b$ are points of increase of $K^{(n)}(x,dy)$. Thus for $n \geq N$, every interval I of length $\geq \epsilon > h$ within $[0,b]$ contains at least one point of increase of $K^{(n)}(x,dy)$, and has $K^{(n)}(x,I) > 0$, as required.

To extend the proof to M+1 machines, take some given $n_1,\ldots,n_M$, $y_1,\ldots y_M$, and $u_1 \ldots u_M$ such that u_i is a point of increase of $K^{(n_i)}(y_i,dy)$, for the two machine transition kernel. Then for the M+1 machine transition kernel $K(\underline{x},d\underline{z})$, $(u_1, u_1+u_2,\ldots, u_1+u_2+\ldots+u_M)$ is a point of increase for $K^{(n_1+\ldots+n_M)}(y_1,\ y_1+y_2,\ldots,\ y_1+\ldots+y_M,d\underline{z})$. This is shown by induction: Assuming $\underline{z}^{(j-1)} = (u_1,\ u_1+u_2,\ldots,\ u_1+\ldots+u_{j-1},\ u_1+\ldots+u_{j-1}+y_j,\ \ldots,\ u_1+\ldots+u_{j-1}+y_j+\ldots+y_M)$ is a point of increase of $K^{(n_1+\ldots+n_{j-1})}(y_1,y_1+y_2,\ldots,\ y_1+\ldots+y_M,d\underline{z})$. Then the n_j step transition from $\underline{z}^{(j-1)}$ has $\underline{z}^{(j)} = (u_1,\ u_1+u_2,\ldots,\ u_1+\ldots+u_{j-1},\ u_1+\ldots+u_j,\ u_1+\ldots+u_j+y_{j+1},\ldots)$ as a point of increase. This is all that is necessary to go from 2 to M+1 machines. □

<u>Proposition A.4</u>: Propositions A1, A2, A3 imply ergodicity.

<u>Proof</u>: This differs from Feller's theorem 2 (op cit) in that strictly positive is replaced by asymptotically strictly positive, and we prove it for R^M. Let then α denote the stationary measure. Let E denote expectation with respect to α. Let $U_o \in C[-\infty,\infty]$ (continuous with limits at $\pm\infty$, hence uniformly continuous). By stationarity $E(U_o) = E(U_1)\ldots$. Also, $E|U_k|$ decrease with k, since:

$$E|U_k| = \int \left| \int U_{k-1}(y)\ K(x,dy)\ \right| d\alpha(x) \qquad (A.8)$$
$$\le \int \int |U_{k-1}(y)|\ K(x,dy)\ d\alpha(x) = E|U_{k-1}|$$

so $\lim_{k\to\infty} E|U_k| = m$ exists. If U_0 is uniformly continuous this implies that U_k are equicontinuous (proposition A.2) and so (by the selection theorem, see Feller, ditto, chapter VIII section 6 theorem 3) a convergent subsequence $U_{n_k} \to V_0$ exists. Applying the transition N times $U_{n_k+N} \to V_N$. By dominated (or bounded) convergence, $E(U_{n_k}) \to E(V_0)$, and, $E|U_{n_k}| \to E|V_0|$, similarly for V_N, and so:

$$E(V_N) = E(V_0) = E(U_0),\ E|V_N| = E|V_0| = m.$$

We now show that this implies that V_0 cannot change sign. By the definition of the stationary measure, similar to (A.8), $E|V_N| = E|V_0|$ is equivalent to:

$$\int \left| \int V_0(y)\ K^{(N)}(x,dy) \right| \alpha(dx) = \int \int |V_0(y)|\ K^{(N)}(x,dy)\ \alpha(dx). \qquad (A.9)$$

Assume that V_0 does change sign. Then by continuity (V_0 is continuous since $U_n \epsilon C[-\infty,\infty]$) we can find $\epsilon>0$ and two open intervals $I_1, I_2 \subseteq \Omega \cap (0,1/\epsilon)^M$, such that $|I_1|>\epsilon$, $|I_2|>\epsilon$ and $V_0>0$ on I_1, $V_0<0$ on I_2. By asymptotic strict positivity of K, we can find N such that for all $x \in \Omega \cap (0,1/\epsilon)^M$, $K^{(N)}(x,I_1) > 0$ and $K^{(N)}(x,I_2) > 0$. But this contradicts (A.9). In particular, if $E(U_0) = 0$ then $V_0 \equiv 0$, and (by considering $U_0(x)-m$) in general, $V_0(x) \equiv m$. Hence, V_0 is constant, and so are of course $V_1, V_2, \ldots$. This limit is independent of the subsequence, and so $U_n(x) \to E(U_0)$. In other words, if for any x we look at the sequence of distributions $K^{(n)}(x,dy)$, and take expectation, denoting the expectation by E_n, we have $E_n(U_0) \to E(U_0)$ for all $U_0 \in C[-\infty,\infty]$. But this implies $K^{(n)}(x,dy) \to \alpha(dy)$ at all points of continuity of α. □

HEURISTICS FOR STOCHASTIC SCHEDULING AND THE CONTROL OF QUEUEING NETWORKS

A REPORT FOR NSF GRANTEE MEETING
TEMPE, ARIZONA JANUARY 1990

GIDEON WEISS
SCHOOL OF ISYE, GEORGIA TECH, ATLANTA, GA 30332

Abstract
We present some of our own research and related results by other researchers. The emphasis is on how the theory of stochastic scheduling can provide insights and proper perspectives for the practitioner. We present one of the first examples in the literature of a worst case analysis of a heuristic for a stochastic scheduling problem, and introduce the new concept of turnpike optimality. We explore the theme of deriving sophisticated priority indices in scheduling situations of various degrees of complexity. We try to deepen our understanding of the complexities of simultaneous operations on the shop floor.

0. Introduction

Role of scheduling
The scheduling of jobs and the control of their flow through a production process is essential to modern manufacturing. Increasing complexity of these processes, rising capital outlay and greater automation will make it even more crucial.
Three distinct approaches to these problems are: deterministic scheduling theory, queueing networks and expert systems.

Deterministic scheduling:
problems are formulated and solved using the methods of combinatorial optimizaiton, see Conway, Maxwell and Miller [1967], Baker[1974], Lawler et al.[1989] for introduction and survey; for recent significant work see Adams, Balas and Zawack[1988]. The difficulty with this approach is that problems are as a rule too big to be solved satisfactorily, especially since the static formulation necessitates frequent resolving.

Queueing Networks:
Scheduling is modeled as a dynamic and stochastic process, see Kleinrock [1976], Kelly [1979], Harrison [1988], and for recent progress Wein[1989]. The difficulty here is that many features of real problems lead to intractable models, and that the use of queueing networks for optimization rather than for modelling is currently still limited.

Expert systems:
Encompass all software based methods which are used to solve scheduling problems, including proprietary programs such as Optima [ ], generic hierarchical approaches such as Gershwin [1987], Leachman [ ], Roundy [ ], and plant specific programs. These include features such as consideration of data flow and user interaction which are ignored by the other approaches. The challenge here is how to incorporate enough modelling and algorithmic sophistication.

stochastic scheduling:
Is attempting to bridge some of the gaps between the three approaches, and in doing so it can provide some very useful insights. In the current note we present some of our own and others' results to illustrate this point. The emphasis is on heuristics and on approximations. One theme is our attempt to understand the complexity of simultaneous operation, as opposed to the ease in which a single processor can be optimized - we present our understanding of the simplest case of identical machines in parallel, as well as an approach for general networks. Another theme is performance evaluation of algorithms and heuristics in stochastic scheduling - we do some worst case analyses and introduce the concept of turnpike optimality.

1. Single machine nonpreemptive scheduling

We consider jobs which undergo a single processing stage. Job j arrives at time A_j waits for a duration W_j then starts its processing at time S_j, for a processing duration X_j and is completed and departs at time C_j. The flowtime of the system is often defined as the average of $C_j - A_j = W_j + X_j$, the sojourn time of job j in the system. The flowtime is one important measure of a system's performance. It may sometimes be possible to control the arrival times and the processing times of jobs - however we assume here that this is not the case and the only control we have is in the order in which jobs are processed. Minimization of flowtime is therefore equivalent to minimizing the average or the sum of the starting times, the waiting times, or the completion times; we will consider the average waiting time in this section.
A single processor is available to process all the jobs, and jobs are processed with no preemption. We will look at two cases: scheduling a batch of n jobs which are all available at time 0, and scheduling an infinite stream of jobs with Poisson arrivals. In both cases the flowtime is minimized by the SPT (shortest processing time first) priority rule. In a stochastic version of this problem, if X_j are only known stochastically, SEPT (shortest expected processing time first) minimizes the expected flowtime.
While the rule is similar whether the processing times are known precisely or approximately, the expected flowtime is not the same. In this section we give some approximation formulas for the effect of lack of information on the flowtime. The results go back to Conway, Maxwell and Miller[1967], to Cobham[1954] and to Schrage and Miller[1968] but they have been somewhat sharpened recently by Matloff[1988] and by Smith and Weiss[1988].

1.1 Batch jobs

We shall assume the following framework - the processing times of the n jobs, $X_1,\dots,X_n$, are drawn randomly from a population with distribution F, let m_1 and σ denote the mean and standard deviation of F. This assumption can often be made, even in deterministic scheduling, and in practice one usually has a fairly good idea about F which summarizes what jobs the system handles as a rule. The scheduler has some information about the jobs, from which he can predict the jobs duration as $Y_j = E(X_j|\text{information})$. We assume that $Y_1,\dots,Y_n$ are again a random sample, from a distribution G with smaller variance than F (this assumption means that we have the same kind of random information on all the jobs). Using this information the scheduler sequences the jobs optimally from smallest to largest Y_j. Taking expectation of the average waiting time, first conditionally on $Y_1,\dots,Y_n$ and then over $Y_1,\dots,Y_n$, one gets the following approximation formula:

$$E(\text{Average wait}) \cong \tfrac{1}{2}\,(n-1)\; m_1\left(1 - r\, d\, \frac{\sigma}{m_1}\right)$$

where:
σ/m_1 is the coefficient of variability of F,
d is a shape parameter of the distribution, defined by
$$m_1 - d\,\sigma = m_{1,2} = E(\min(X_1,X_2))$$
that is $m_{1,2}$ is the expected value of the smaller of two independent observations from F. Typically, $d \cong \frac{1}{2}$ (for the exponential distribution - $d = \frac{1}{2}$, for the normal distribution - $d = 0.5642$, for the uniform distribution - $d = 0.5774$).
r is the correlation coefficient between the processing time X_j and the predicted processing time Y_j.

The two special cases of $r = 1$ and of $r = 0$ correspond to exact knowledge and to no knowledge at all about $X_1,\dots,X_n$; In those two cases the formula is exact. For $0 < r < 1$ the formula is only exact under linear model assumptions - one case of a linear model is when processing times are normally distributed.
To examine the formula, we note that $0 \le r \le 1$, $d \cong \frac{1}{2}$, and (for positive random variables) usually $\sigma/m_1 \le 1$. Hence, the ratio of average waiting time per job without information and with information is as a rule ≤ 2; the value 2 is obtained for instance for exponentially distributed processing times, when comparing full information to no information at all. We note that 2 is not an upper bound and examples of arbitrarily high ratios can be constructed, by using extremely over dispersed F.

1.2 stream of jobs

An infinite stream of jobs are arriving at times $0 \le A_1 \le A_2 \le \cdots$, and require processing times $X_1, X_2, \cdots$. We assume that $A_1, A_2, \cdots$ form a Poisson process with rate λ, and that the $X_1, X_2, \cdots$ are independent identically distributed from the processing time distribution F, with mean m_1. Another parameter of F is relevant here, namely $B = \text{esssup}(F) \le \infty$; loosely speaking, B is the longest processing time which jobs drawn from F are ever likely to have. We assume that the arrival rate λ is smaller than the processing rate $\mu = 1/m_1$ so that the traffic intensity $\rho = \lambda/\mu$ is less than 1, and therefore the process is stable. We are interested in the average waiting time per job (as surrogate to flowtime). We note that in a more deterministic world, if arrivals are equally spaced and all service times are equal there is no waiting at all so long as $\rho \le 1$, so unlike the batch case, the waiting here is purely the result of stochastic variability.
We again present an approximation formula for the effect of partial information on the average waiting time. This formula is based on heavy traffic arguments and holds as $\rho \to 1$, when the average number of jobs queueing for the processor and the average waiting time are very large.
Full information in this case means that each job's processing time becomes known upon his arrival. For a Poisson arrival stream the optimal

policy is to start the shortest waiting job whenever the processor becomes available, that is the SPT rule. With partial information, predicted processing times replace the unknown processing times, that is SEPT is optimal. With no information, all policies lead to the same average waiting time, though policies may differ on other criteria. For instance, FIFO (first in first out) will give a smaller waiting time variance than LIFO (last in first out).

The heavy traffic formula is

$$\frac{E(\text{waiting time}|\ SPT)}{E(\text{waiting time}|\ FIFO)} = \frac{B}{m_1}$$

An intuitive argument to derive this formula is as follows: by work conservation (that is no unnecessary idling of the processor), the amount of work in the system at any time t is independent of the policy, let S be the long time temporal average amount of work in the system. If no information is available on the job processing times then this amount represents an average of $L(FIFO) = S/m_1$ jobs in the queue. If however information on processing times is available and long jobs have lower priority than shorter jobs, then in heavy traffic almost all the jobs queueing up will be B long, so the workload S represents only $L(SPT) = S/B$ waiting jobs. Applying Little's formula which says that $L = \lambda W$ where W is the average waiting time of a job, yields the formula.

The important thing to note here is that the ratio can be much higher than in the batch case, where ≤ 2 was typical; indeed, if job processing times are not essentially bounded, so that $B=\infty$, the ratio can be arbitrarily large as $\rho \to 1$. Of course one pays for the gain in average flowtime by extremely long waiting times of long jobs.

2. Single machine preemptive scheduling

We consider the batch scheduling problem of section 1.1, but we now allow preemption of jobs. In the deterministic case, when all processing times are known in advance, it never pays to use preemptions - if a job needs to be preempted it should not have been started. In the stochastic case, when the processing time of a job becomes known only as it is being processed, we can make use of this dynamic information, and improve the flowtime by using preemptions. It turns out that there is a priority index which we can calculate for each job, and which changes dynamically as the job is processed. We describe this index and the resulting optimal policy next. This priority index is an example of a Gittins dynamic allocation index, which is defined for so called bandit processes, and solves the multiarmed bandit problem. We discuss the Gittins index in section 2.2, and return to it in later sections. For references on the scheduling problem see Sevcik[1974], Harrison[1975], Meilijson and Weiss[1975], for the bandit problems and the Gittins index see Gittins[1979,1989], and Whittle[1980].

2.1 Priority index and optimal preemption of jobs

Consider a job whose processing time X is drawn from a distribution F. Assume we have processed this job for a duration x and the job has not yet been completed; we call x the age of the job. The information which we obtained by watching the job for a duration x and knowing that it is not yet complete is summarized by the conditional distribution $F(\cdot|X>x)$. We may now decide to give this job an additional period of processing of up to y time units if required - of course the job may finish before using all of y. Define the preindex

$$v(x,y) = \frac{\int_x^{x+y} dF(t)}{\int_x^{x+y}(1-F(t))dt} = \frac{P(\text{completion in } (x,x+y))}{E(\text{time used in } (x,x+y))}$$

and the index for the job at age x

$$v(x) = \max_{y>0} v(x,y)$$

In the batch scheduling problem, assume there are n jobs, with processing time distributions $F_1,\dots,F_n$, and current ages $x_1,\dots,x_n$. Let $v_1(x_1),\dots,v_n(x_n)$ denote the current values of their indices. Then

Theorem: To minimize expected flowtime it is optimal to always schedule the job with the highest current index.

2.2 Bandit processes and the Gittins Index

Consider n processes with states at time t, $t = 0,1,\cdots$

$$X_1(t), \quad \cdot \quad \cdot \quad \cdot \ , \quad X_n(t)$$

at any time t, exactly one of the processes, say $i(t)=i$, is made active; all other processes are passive. The active process undergoes a state transition according to Markov transition probabilities

$$P_i(x,y) = P(X_i(t+1)=y|X_i(t)=x)$$

while the passive states are frozen - $X_j(t+1)=X_j(t)$, $j\neq i$.

A reward $R(T) = R_i(X_i(t))$ is earned by the active process, the passive processes earn nothing.

This system of n processes is called by Gittins a system of alternative bandit processes. The problem is to decide at each time point which process to activate, so as to maximize $\Sigma\beta^t R(t)$, where $\beta \leq 1$ is a discount factor.

Gittins has defined his index for each of the processes as

$$v_i(x) = \max_{\tau>0} \frac{E(\sum_{t=0}^{\tau-1}\beta^t R_i(X_i(t))\ |X_i(0)=x)}{E(\sum_{t=0}^{\tau-1}\beta^t\ |X_i(0)=x)}$$

where the maximization is taken over all possible stopping times (note that a stopping time τ is as a rule random - it can depend on where the process $X_i(t)$ will go, for example, it may be the first time that the process enters a certain set of states). In words, the index expresses the maximal reward per unit time (both discounted) that can be achieved by the process, starting at state x, and choosing the best amount of time to stay active. Gittins has shown:

Theorem (Gittins): The optimal policy is at any time to activate the bandit process with the highest index.

The importance of Gittins' result is that the indices are calculated for each process separately, and do not depend on the other $n-1$ processes. The n dimensional problem of what to do in state $X_1(t),\dots,X_n(t)$ is uncoupled to solutions of n one dimensional problems.

Gittins result solved Bellman's multiarmed bandit problem. As we can see it also applies to section 2.1, where the index of a job is an example of a Gittins index (with discount factor 1). It also solves Klimov's problem of the control of the M/G/1 queue and its extensions, as discussed in section 5.

Whittle has given an equivalent formulation of the Gittins index, in terms of a "retirement reward". For process i consider the problem of choosing at each point in time either to be active and collect the usual reward, or to retire and collect a retirement reward v. This is a Markov decision problem with optimality equation:

$$V(x,v) = \max(R_i(x) + \Sigma P_i(x,y)V(y,v),\ v)$$

Whittle shows that the value v for which both actions are optimal when process i is in state x in the above problem, is $v_i(x)$. This alternative formulation is the starting point for restless bandits discussed in section 6.

3. machines in parallel

In this section we look at the simplest situation when there is simultaneous processing of more than one job, the operation of $M+1$ identical machines which are working in parallel. We consider a batch of n jobs, each of which can be done on any of the machines. In addition to flowtime, which we express as ΣC_j, we will consider minimization of two other objective functions: weighted flowtime $\Sigma a_j C_j$, where a_j, the weight of job j, can be thought of as holding cost per unit time, and makespan $C_{max} = \max(C_j)$, the time needed from start to completion of the last job.

These problems become hard on parallel machines even in the deterministic case. In the following discussion we believe we show that analysis of the stochastic problems provides an understanding of parallel operation and a more realistic evaluation of heurisitcs.

3.1 Deterministic scheduling of parallel machines.

Flowtime on parallel machines is minimized by SPT as on a single machine.

Minimization of makespan (which for a single machine is independent of the schedule) on parallel machines is an NP-hard combinatorial optimization problem. However there is little doubt that it is an extremely easy problem to solve satisfactorily: if n is small it can be solved enumeratively; for n moderate, polynomial approximation schemes exist; for n large, LPT is an excellent heuristic; in fact, for large n every non idling schedule will be almost optimal in practice, since the only optimization involved is how to make the $M+1$ last jobs finish together, but the effect of $M+1$ jobs is usually negligible for large n. See Karmarkar and Karp [1982], and Frenk and Rinnooy Kan [1987] for details, we say no more on makespan here.

Minimization of weighted flowtime on a single machine is achieved by Smith's rule: start the jobs in decreasing order of a_j/X_j. The problem for $M+1$ parallel machines is however NP-hard. Smith's rule is a natural heuristic for this problem. Intuitively, what it does is to reduce the holding cost rate of the remaining jobs fastest, for one machine as well as for $M+1$ machines, except for the end of the schedule, when machines fall idle as the last $M+1$ jobs are completed, and processing becomes inefficient. This end effect is the source of the difficulty, as it is also for the makespan problem. Worst case analysis shows that the SR (Smith's rule) heuristic can be 1.20 times worse than the optimal solution, see Kawaguchi and Kyan [1986].

3.2 Optimality results for stochastic scheduling on parallel machines

The most important exact optimality result for parallel machines is the following:

Theorem (Weber, Varaiya and walrand [1986]): If the processing times are stochastically comparable random variables than SEPT minimizes the expected flowtime.

We say X and Y are stochastically comparable if $X \leq_{ST} Y$ or $Y \leq_{ST} X$; we say X is stochastically smaller than Y, $X \leq_{ST} Y$, if $P(X>t) \leq P(Y>t)$ for all t.

This theorem subsumes several earlier results discovered for specific distributions.

No general optimality results exist for the stochastic problem of minimizing weighted flowtime. If Smith's rule agrees with SEPT than it is optimal for exponentially distributed processing times; see Weiss and Pinedo[1980], Kämpke[1987].

LEPT (longest expected processing time first) minimizes the expected makespan if the processing times are exponentially distributed, and for a few other special cases Weiss and Pinedo [1980], Weber [1982].

Limited as these results are, they seem to be fairly close to exhaustive - we doubt whether there is much more to be discovered here.

3.3 Formulas for flowtime of machines in parallel

Sections 3.3-5 are a summary of Weiss[1988a,1988b].

3.3.1 Expected flowtime for iid jobs.

Consider a stream of jobs with processing times $X_1, X_2, \cdots$, iid drawn from a distribution F; let m_1, σ^2, and m_3 denote the mean, variance and third moment of F. A set of $M+1$ machines become available to process these jobs at times $U_{00} \le U_{10} \le \cdots \le U_{M0}$, assume $\Sigma U_{i0} = 0$. The stream of jobs is processed on the $M+1$ machines in parallel, and so the completion times of jobs on the different machines form $M+1$ independent renewal processes. Let $U_{0n} \le U_{1n} \le \cdots \le U_{Mn}$ denote the times at which the $M+1$ machines complete the first n jobs, let $D_{in} = U_{in} - U_{i-1n}$, $i=1,\dots,M$, and let

$$S_n = \frac{1}{M} \Sigma\ U_{in}^2 - \frac{1}{M(M+1)} (\Sigma\ U_{in})^2.$$

Theorem (Weiss 1988): As $n \to \infty$, the remaining processing times of the last M jobs among jobs $1,\dots,n$, $D_{1n},\dots,D_{Mn}$, converge in distribution to an ordered sample from the equilibrium distribution of F with probability density function $f_e(t) = (1-F(t))/m_1$, if F is nonarithmetic.

The expected flowtime of n iid jobs on $M+1$ machines follows from this theorem:

$$E(\Sigma C_j) = \frac{n(n+1)}{2(M+1)} m1 + \frac{n\ M}{2(M+1)} m_1 (1 - \frac{\sigma^2}{m_1^2}) - \frac{M}{2} \frac{S_0^2}{m_1} + \frac{M}{2} \frac{E(S_n^2)}{m_1} \qquad (1)$$

Each of the four summands in this formula has a meaning: the first is equal to the expected flowtime on a single machine with $M+1$ fold speed - that is a single machine that has the combined capacity of the $M+1$ parallel machines; the second accounts for a constant delay per job that is the result of parallel processing - this delay equals the difference between the first moments of F and of f_e, we call it a synchronization delay; the third and fourth are contributions of starting conditions and ending conditions. As $n \to \infty$, the fourth summand

$$\frac{M}{2} \frac{E(S_n^2)}{m_1} \to \frac{M}{M+1} \frac{m_3}{3 m_1^2} + \frac{M-1}{M+1} \frac{m_1}{4} (1 - \frac{\sigma^2}{m_1^2})^2 .$$

If one adds and subtracts this limit from the third and fourth summand the formulas' four parts have orders of magnitude $O(n^2)$, $O(n)$, $O(1)$, and $o(1)$.

Some specific examples of this formula are:

(1) Constant processing times $X_j = 1$. Here we assume that starting times of machines are staggered, with intervals $1/M+1$:

$$E(\Sigma C_j) = \frac{n(n+M+1)}{2(M+1)}$$

(2) Exponential processing times, mean 1, all machines start at 0:

$$E(\Sigma C_j) = \frac{n(n+1)}{2(M+1)} + \frac{M}{2}$$

(3) Uniformly distributed processing times, on (0,2), all machines start at 0.

$$E(\Sigma C_j) = \frac{n(n+\frac{2}{3}M+1)}{2(M+1)} + \frac{M(M+2)}{9(M+1)}$$

(4) Decreasing hazard rate jobs - these are jobs which as long as they are not completed, the longer you work on them, the longer their remaining processing time becomes. this is perhaps typical of R&D activities and not of manufacturing. Such jobs have $\sigma/m_1 > 1$, and so the synchronization delay is negative - in other words, it is best to process them in parallel and not one after the other on a single machine.

3.3.2 Expected weighted flowtime

Formula (1) exhibits a large degree of insensitivity to the assumption of iid processing times. If all jobs have equal first and second moments, no change is necessary; if the means are equal and variances differ, replace σ^2 by $\Sigma \sigma_j^2/n$. More generally, we can get an expression for the expected weighted flowtime in the following special case

Theorem (Weiss 1988): Let the processing times of the n jobs be independent random variables X_j with means $E(X_j) = 1/\mu_j$, variances σ_j^2, and assume that the flowtime (completion time) of job j has weight $1/\mu_j$, for $j=1,\dots,n$. Then

$$E\left(\Sigma C_j/\mu_j\right) = \frac{1}{2(M+1)} \left(\Sigma\, 1/\mu_j\right)^2 + \frac{1}{2} \Sigma \left(1/\mu_j\right)^2 - \frac{M}{2(M+1)} \Sigma\ \sigma_j^2$$
$$- \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2).$$

Two things are remarkable about this formula: except for the last term, it depends on the distributions of the processing times only through their means and variances; and, more surprising, except for the last term, it is completely independent of the schedule.

The essence of the following results is the way in which this formula pushes all the complications to the "end effects" term.

3.4 Approximate optimality

We now present a bound on the difference between expected weighted flowtime under Smith's rule, and under any other schedule. We consider n jobs, with expected processing times $1/\mu_j$ and with weights a_j, $j=1,\dots,n$. Define

$$D^2 = \max_{1 \le j \le n} \sup_{s} E((X_j - s)^2 \mid X_j > s)$$

In words, D^2 measures how large the square of the remaining processing time of an unfinished job is expected to be, at the most.

Theorem (Weiss 1988): Let SR denote the use of Smith rule, and let Π be an arbitrary nonpreemptive scheduling strategy, then

$$E(\Sigma a_j C_j | SR) - E(\Sigma a_j C_j | \Pi) \le \frac{M^2}{2(M+1)} \max_{1 \le j \le n} (a_j \mu_j)\ D^2 \qquad (2)$$

This theorem holds for strategies Π which allow the use of inserted idle time, and it holds in the case where the actual processing time of each job becomes known at the instant at which it starts being processed, and this knowledge can be used by Π.

We now assume that all $a_j \mu_j$ and D^2 are bounded by a constant, independent of the number of jobs n. There are many practical examples when this assumption holds. Note that this is only an assumption on the distributions of processing times, and it does not mean that the actual processing times are bounded in any way. Under this assumption, the worst case ratio is

$$\frac{E(\Sigma a_j C_j | SR)}{E(\Sigma a_j C_j | \Pi)} \cong 1 + O(\frac{1}{n^2})$$

which is very different from the deterministic worst case ratio of 1.20.

3.5 Turnpike optimality

In the last section we saw that the expected value of the objective function under Smith's rule is close to optimal. We now show that most of the time the optimal action is to follow Smith's rule. We call this the "turnpike optimality" property of Smith's rule in analogy with turnpike optimality in optimal control problems and in discounted dynamic programming. Turnpike optimality has to the best of our knowledge not been considered previously in scheduling theory; we believe it is of great practical importance.

In the context of last section define

$$\delta^{(M)} = \min_{j_0 \cdots j_M} \inf_{s_0 \cdots s_M} E(\min(X_{j_0} - s_0, \cdots, X_{j_M} - s_M) \mid X_{j_0} > s_0, \cdots, X_{j_M} > s_M)$$

In words, $\delta^{(M)}$ measures how small in expectation can the interval be from a time point when all $M+1$ machines are occupied by unfinished jobs, until one of the machines becomes available to start a new job. If the processing times have discrete probability distributions, then $\delta^{(M)}$ will simply be the unit of the discrete distributions. Let also:

$$\mu_{max} = \max (\mu_j)$$
$$(a\mu)_{max} = \max (a_j \mu_j)$$
$$(\Delta a\mu)_{min} = \min |\ a_j \mu_j - a_k \mu_k\ |$$

Theorem (Weiss 1988): Let Π_0 be a nonpreemptive work conserving nonpredictive and nonrandomizing strategy. Let L be the number of times that Π_0 starts a job not according to SR. Then if

$$E(\Sigma a_j C_j | SR) - E(\Sigma a_j C_j | \Pi_0) \ge 0$$

that is Π_0 is better than SR, then

$$E(L | \Pi_0) \le \frac{M^2}{2} (a\mu)_{max}\ D^2\ \mu_{max} / (\Delta a\mu)_{min}\ \delta^{(M)}$$

Again under reasonable assumptions this bound does not grow with n and so as $n \to \infty$,

$$E(L | \Pi_0) / n \to 0.$$

4. machines in series

Following machines in parallel, we now turn to machines in series (in tandem is the term used in queueing theory), or flowshops. In a flowshop of M machines, ordered $1,\dots,M$, each job needs to go through the machines in that order. We will consider permutation flowshops, in which the jobs are started on each of the machines in the same order - there seems to be no advantage to doing anything else. Note that even though each job sees the machines as a series, the M machines of the flowshop are actually working simultaneously. The flowshop is much more complicated than machines in parallel, and we are much further from understanding it. In queueing theory tandem queues were investigated by Tembe and Wolfe[1974], Greenberg and Wolfe[1988], Whitt[1985], and Wein[1988]. A very interesting paper with a lot of insight is Harrison and Wein[1988]. We make no attempt to discuss all this work here or to give an exhaustive survey. Our discussion is of two topics: probabilistic analysis of the two machine flowshop, using Johnson's rule and other rules; and choosing the order of machines in a flowshop with blocking.

4.1 Two machine flowshop

In a batch of n jobs, job j requires A_j processing on machine 1, B_j processing on machine 2. We assume that $A_1,\dots,A_n$ are drawn iid from a distribution F_A, with mean m_A, and $B_1,\dots,B_n$ are drawn independent of $A_1,\dots,A_n$ and iid from a distribution F_B, with mean m_B. These distributional assumptions enable us to do a probabilistic analysis of the system. We assume however that the actual values of $A_1,\dots,A_n$ and $B_1,\dots,B_n$ are known to the scheduler, so we are analyzing a deterministic problems. Johnson's rule says: divide the jobs into two sets - those with $A_j \le B_j$ are scheduled first, according to SPT order of the A_j, those with $A_j > B_j$ are scheduled next, according to LPT order of the B_j.

Johnson's rule is the rule which minimizes the makespan of all the jobs. It achieves that by causing the least possible idling on machine 2. The problem of minimizing flowtime (and indeed almost every other problem on flowshops) is NP-hard. Perhaps because it is the only rule with some positive distinction, Johnson's rule is quite popular. Yet the following analysis shows that there is very little merit to the use of that rule. Consider first the case $m_A < m_B$. Then it is enough to do just a few jobs according to Johnson's rule and proceed arbitrarily thereafter, and with very high probability there will be no idling at all on machine 2. and the policy will achieve the makespan of Johnson's rule. In fact, for moderately large n, the makespan of a random order of jobs is $n\, m_A + O(1)$, and Johnson's rule is superfluous. The case $m_B < m_A$ is the same by reversibility (see Yamazaki and Sakasegawa [1975], Muth [1979]).

In the case $m_A = m_B = m$, we assume for simplicity $\sigma_A = \sigma_B = \sigma$. For moderately large n, the expected makespan is bounded below by $n\,m + 0.5642 \sqrt{n}\, \sigma + O(1)$. Johnson's rule achieves this bound up to an $O(1)$ term. However, to do so it builds up a queue of approximately $n/4$ jobs between the two machines, which persists for more than half the makespan. Some simple heuristics can achieve the same makespan (up to order $O(1)$), with a queue that is of order magnitude $\sqrt{n}$. A random schedule gives a makespan of $n\,m + c\sqrt{n}\,\sigma$.

4.2 Optimal order of machines with blocking

We consider M machines in series with an infinite supply of jobs in front of the first machine, and no waiting room between the machines. The jobs are assumed to require iid processing times. A typical job requires processing for durations $X_1,\dots,X_M$ on the M machines, which are assumed independent from distributions $F_1,\dots,F_M$. Jobs are fed into the system whenever the first machine is free. they move through the M machines and out. Because there is no waiting room there is no queueing but jobs may be blocked in one machine, after processing is complete, because the nest machine is not free. The problem is to choose the right order of machines, or in other words the right permutation of $F_1,\dots,F_M$.
This problem has been studied by Yamazaki et al [1989], with the objective of maximizing the throughput of the system. The problem is very hard - all that can be proved is that under suitable conditions one should have slower machines in the first and last position than in the second and penultimate positions respectively (Huang and Weiss [1989], Shantikumar et al [1989]). It is clear that this problem does not possess a "nice" solution beyond the rule of thumb that "slow machines should be kept as separate as possible".
However another objective here might be to minimize the average time spent by a job from entering the first machine until it leaves the last, that is the average wait or flowtime. We conjecture that for this problem, machines should be ordered from slowest first to fastest last.

5. Queueing networks with a single server

In section 1.2 we considered jobs that arrive in a Poisson stream at a single server queue, and are processed with no preemptions; scheduling waiting jobs in this system according to SEPT minimizes the expected flowtime (in the form of the average waiting or sojourn time). If jobs incur different holding costs then scheduling them according to Smith's rule, or , as it is called in the queueing literature the "$c\mu$ rule", minimizes the expected weighted flowtime. For these two problems job j has $E(X_j)$ and $a_jE(X_j)$ as priority index, and the optimal policy is a priority policy - at any decision time schedule the job with the highest priority. In section 2.1 we considered scheduling a batch of jobs, when preemptions were allowed: We found that for job j of age x_j we could calculate a priority index $v_j(x_j)$, and the priority policy based on this priority index minimizes the expected flowtime for the batch. It turns out that the same priority policy minimizes expected flowtime also for a Poisson stream of arriving jobs, and that the weighted flowtime problem has $a_jv_j(x_j)$ as the appropriate index. In this section we generalize these models further. We start with Klimov's model on control of the M/G/1 queue, and we then present even more general branching bandit processes. One view of this general version is to regard it as a queueing network, with a single server that is moving around the nodes of the network. All these problems are solved optimally by a priority policy. The priority index for this policy is a Gittins type priority index, and the optimality results are generalizations of Gittins original results.

5.1 Klimov's model for the control of the M/G/1 queue.

Jobs of several types, denoted by k, $k=1,\dots,K$, arrive at a single server station, in independent Poisson arrival streams, type k have arrival rate λ_k, $\lambda_k \geq 0$. A job of type k requires processing time of duration X_k, with mean m_k, all processing times are assumed to be independent. Jobs cannot be preempted while in process. On completing his processing, a job of type k may rejoin the queue, with probability P_{kk}, or rejoin the queue as a job of a different type, say type j, with probability P_{kj}, or it may leave the system, with probability $1 - \Sigma_j P_{kj}$. A holding cost of a_k per unit time is charged to any type k job in the system. The problem is how to schedule jobs for processing, so as to minimize the long run average holding costs (this is equivalent to minimizing weighted flowtime). Klimov[1974] solved this problem, using queueing theory techniques; Sevcik[1974], Harrison[1975], Meilijson and Weiss[1977], Tcha and Pliska[1977] and Nash and Gittins [1977], and more recently Varaiya Walrand and Buyokoc[1985] and Weiss [1988] contain various proofs and generalizations, using Markov decision processes.
It is easy to see that the problems of sections 1.2 and 2.1 are special cases: In the problem of section 1.2, every customer leaves the system when his service is complete. For the preemptive scheduling problem, assume job j requires processing time X_j which is discrete with $P(X_j=x) = P_j(x)$, and with $P(X_j>x) = Q_j(x)$, $x=0,1,\cdots$. We define (j,x_j), job j at age x_j, as a type - then the service time of this type is 1, and at this service end the job leaves the system (is completed) with probability $P_j(x_j)/Q_j(x_j-1)$ and it turns into type (j,x_j+1) with probability $Q_j(x_j)/Q_j(x_j-1)$.
Whenever a service is completed, the server is faced with a state given by $n_1,\dots,n_K$ where n_k is the number of type j customers in the system, and he needs to decide which of these to serve. At first glance this appears to be a formidable problem because of the large K dimensional state space. In fact the optimal policy is a priority policy, where the types have a priority index and at any state the optimal policy is to start serving one of the customers with the highest priority index among those in the queue. The priority index of a type k job is

$$v(k) = \max_{\tau>0} \frac{E(\text{cost reduction} \mid \text{started as type k})}{E(\ \tau \mid \text{started as type k})}$$

where one works on a single job, starting as a type k job with cost rate a_k and through a possible sequence of type changes until a stopping time τ, at which point if the job left the system the cost reduction is a_k, if the job is now a type j the cost reduction is $a_k - a_j$. This is clearly a Gittins index.
The above definition appears unsuitable for calculation. The following recursive procedure calculates the indices of types $1,\dots,K$:
step 1 - Determining the highest priority type: For each type k calculate the expected cost rate reached at the end of X_k, call it $d_1(k)$, and calculate the ratio $(a_k - d_1(k)) / m_k$. Rename the type with maximal ratio type 1, it is the highest priority type and the ratio is its index.
step r - Determining the r'th priority type: Assume types $1,\dots,r\text{-}1$ have been determined as top priority types. For type k, $k \geq r$, do: Calculate the expected time it takes for a type k customer until it leaves the system or until it changes into a customer of one of the types $r,r+1,\dots,K$, call this expected time $\eta_r(k)$. Calculate the expected cost rate reached at that time, call it $d_r(k)$. Calculate the ratio $(a_k - d_r(k)) / \eta_r(k)$. Rename the type with maximal ratio type r, it is the r'th priority type and the ratio is its index.
These calculations are easily performed on the partitioned vector of cost rates, vector of mean processing times, and transition probability matrix. It is interesting to note that the arrival rates play no role in the index.

5.2 Branching bandit processes

The following problem formulation generalizes Klimov's setup considerably and recasts it in the form of bandit processes. We have a collection of bandit processes, as in 2.2. For simplicity we assume that these processes are iid, and that they move on a finite state space, with states $1,\dots,K$. The state of the whole system can then be described at any time by $n_1,\dots,n_K$ the number of arms in each of the states. At a decision moment an arm is chosen and made active,while all the other arms are passive. Say the active arm is in state k, then its activation (processing or any other interpretation) will last for a random duration X_k with distribution F_k, at the end of which a reward R_k will be earned (fixed or random), and the arm will be replaced by a new set of "descendant" arms, a random vector of arms, $N_{k1},\dots,N_{kK}$, so that N_{kj} is the number of descendant arms of type j. The passive arms remain frozen and earn no rewards. A discount factor β is used to discount the rewards. It is desired to find a policy which will maximize the discounted sum of rewards. This problem was studied in Weiss[1988] where it is shown that priority order of the states exists, and can be calculated recursively, and the priority policy is optimal.
The descendants mechanism of branching bandit processes enables the modelling of arrival processes more general than Klimov's model, such as batch Poisson arrivals, as well as control of arrivals by the scheduler, also it allows a job to split into several jobs.
With a discount factor <1, the priority order of jobs is no longer independent of the arrival rates. If the discount rate is 1, one can add or eliminate Poisson streams of arrivals without changing the priority order.

5.3 Queueing networks with a single server.

A standard formulation of a queueing network scheduling problem, which is used to model flow of jobs in a factory, is as follows: The network consists of a set of nodes which may be grouped in subsets of nodes that one can think of as service stations. There are several types of jobs which arrive from outside. Each type has a specified entry node and it then follows type specific Markov routing through the nodes until it leaves the network. The sojourn time at each node is also a type specific random variable. One can now take each type-node combination and call these classes. A job of class k will be processed (at a specific node) for a random duration X_k with mean m_k and variance σ_k^2, and on completion of this time it will become a class j job with a probability P_{kj} and leave the system with a probability $1 - \Sigma_j P_{kj}$. Type or class specific holding costs are denoted by a_k.
In this form the problem looks very similar to the Klimov model. Indeed, if we make the (totally unrealistic) assumption that there is only a single operator in the whole factory and he moves about from node to node and can chose at any time which class to serve, while everything else is frozen, then we are back exactly to Klimov's model. In that case we know that there exists a priority order of the classes, and the single operator will always serve the top priority class jobs present in the system. To add a degree of realism to this we can endow this server with a service speed which is the combined speed of all the servers at all the nodes (as we did in section 3.3 when we compared M+1 parallel machines to an M+1 fold speed single machine). Unfortunately it is not at all clear whether this single server network and its Klimov model solution is in any sense an approximation to the multinode multiserver queueing network.
In reality the difficulties are first that service capacities at different nodes or service stations cannot be utilized anywhere else in the system, and second that at a single node when several jobs are present one cannot concentrate all the service capacity at this node (say 5 identical machines) on a single high priority job. Thus at each node we need to activate several jobs, as many as we have machines. Furthermore, while we allocate machines at one specific node, the changes in state at that node are also affected by similar activities at the other nodes.
In the next section we consider restless bandits, where we activate more than one arm (analogous to several machines at a node), and where the states of passive arms are not frozen (perhaps related to activities at other nodes).

6. Restless bandits

6.1 Formulation of restless bandits

Recall the formulation of bandit processes in section 2.2. Consider again n processes with states at time t, $t = 0,1,...$

$$X_1(t), \quad \cdot \quad \cdot \quad \cdot, \quad X_n(t)$$

At any time t, exactly m of these processes, say $i_1(1), i_2(1), ..., i_m(1)$ are made active; all remaining n-m processes remain passive. Active process i undergoes state transition according to Markov transition probabilities

$$P_i^A(x,y) = P^A(X_i(t+1)=y|X_i(t)=x)$$

and collects a reward $R_i^A(X_i(t))$.
At the same time, passive process j undergoes state transition, according to passive transition probabilities:

$$P_i^P(x,y) = P^P(X_i(t+1)=y|X_i(t)=x)$$

and collects passive reward $R_i^P(X_i(t))$.
The total reward at time t is the sum of the rewards of the m active and n-m passive processes. The problem is to operate the system so as to maximize the long term average expected reward per unit time. We denote this reward by $R_{OPT}(m,n)$.
In what follows we will for simplicity assume (as we did in 5.2) that all the processes are identically distributed, and that they move on a finite set of states, 1,...,K. We can then describe the system state by the number of processes in each of the states 1,...,K, as $n_1(t),...,n_K(t)$, or by $z_n(t) = z_{1n}(t),...,z_{Kn}(t)$, where $z_{kn}(t) = n_k(t)/n$.
Whittle [1988] introduced this model, and named it restless bandits because passive processes are not frozen. We gave some motivation for this type of model in the last section; Whittle lists some other applications. Unfortunately, the fact that m processes are activated simultaneously, and a-fortiori the restlessness of the passive processes, make this problem intractable. It is no longer true that a priority policy is optimal, the solution has to be sought in the full multidimensional space, and there are no indications that the solution might have any "nice" features.

6.2 Whittle's relaxed problem and index policy.

If one attempts to maximize the average reward from a single process, one obtains the optimality equation

$$\gamma + h(k) = \max(R^A(k) + \Sigma P^A(k,j)h(j),\ R^P(k) + \Sigma P^P(k,j)h(j))$$

the solution to which will optimally partition the states 1,...,K into active and passive and give an optimal average reward per unit time of γ.
Introduce now a "subsidy" for being passive of magnitude v, then

$$\gamma(v) + h(k) = \max(R^A(k) + \Sigma P^A(k,j)h(j),\ v + R^P(k) + \Sigma P^P(k,j)h(j)) \quad (3)$$

where the optimal average reward $\gamma(v)$ and the partition of the states are now a function of the subsidy v.
Whittle now introduces a relaxed problem: instead of having exactly m active processes at all times, allow any number m(t) of active processes at time t, but require that the long term average of m(t) be m.
To solve the relaxed problem the subsidy v is treated like a Lagrange multiplier for the relaxed constraint, and the optimal reward per unit time for the relaxed problem is:

$$R_{REL}(m,n) = \inf_v (n\,\gamma(v) - v\ (n-m)) \quad (4)$$

Let v^* be the value which minimizes (4). When (3) is solved with v^* a partition of the states into active states, passive states, and one state which is randomized, passive with probability θ, active with probability $1-\theta$, is obtained.
The relaxed policy operates each arm independently of all others, and makes it active or passive according to this partition.
The subsidies can also be used to calculate an index for each state. for state k define the index v(k) as the smallest value of v such that in solving (3) with that value, in state k one is indifferent between active and passive. Note the similarity with Whittle's definition of the Gittins index in section 2.2, and the analogy between "retirement reward" there and "subsidy" here.
The index policy is at any time t to activate the m processes with the highest indices. Denote by $R_{IND}(m,n)$ its average reward per unit time. clearly,

$$R_{IND}(m,n) \le R_{OPT}(m,n) \le R_{REL}(m,n)$$

6.3 Whittle's conjecture on asymptotic optimality

In his paper Whittle conjectured that as $n\to\infty$, the three policies converge to the same reward per unit time per arm, that is

$$\lim_{n\to\infty} R_{IND}(m,n)/n = \lim_{n\to\infty} R_{OPT}(m,n)/n = \lim_{n\to\infty} R_{REL}(m,n)/n$$

This problem was investigated by Weber and Weiss[1989], for iid processes on a finite state space with the following results:
Theorem (Weber and Weiss 1989): Let $\alpha = m/n$, and let $r(\alpha) = R_{REL}(m,n)/n$ (note, this is independent of n). Then

$$\lim_{n\to\infty} R_{OPT}(m,n)/n = r(\alpha).$$

A sufficient condition for the second half of the conjecture is
Theorem (Weber and Weiss 1989): If the problem is indexable, and if the fluid approximation of the system under the index policy is globally asymptotically stable, then

$$\lim_{n\to\infty} R_{IND}(m,n)/n = r(\alpha)$$

The definition of indexability is: let v in (3) vary from $-\infty$ to ∞. At $v = -\infty$ the partition is to have all the states active; at $v = \infty$ the partition is to have all states passive. The problem is indexable if the change in the partitions is monotone throughout.
The proof of the second theorem is based on studying the fluid approximations of the system under the relaxed policy and under the index policy. Under the sufficient conditions in the theorem these two fluid approximations converge to the same point. Some large deviation theorems are then used to show that the stochastic system stays close to this asymptotic point as n becomes large.
Weber and Weiss construct a counterexample to the conjecture in general. However counterexamples of this kind seem very rare, and the asymptotic gap that they exhibit is extremely small.
In conclusion it seems that the index policy combined with the upper bound of the relaxed policy provide an almost picture perfect heuristic for the restless bandit problem.

7. Fluid approximations

Fluid approximations to queues, queueing networks or stochastic systems replace the stochastic systems by a deterministic trajectory which describes the motion of the mean of the stochastic system. Thus they agree with the stochastic system on first moments only. Fluid approximation are discussed in Kleinrock[1979] and in Newell [1982]. Mitra and A. Weiss[1988] have used them to analyze communication systems and derived large deviation results. Chen and Mandelbaum [1987] have investigated convergence of queueing networks to their fluid approximations. Recently Chen [1987] and Chen and Yao [1989] investigated optimal control of fluid networks. The proof of Whittle's conjecture in the last section is based on fluid approximations.
The following scheme seems natural: approximate a network scheduling problem by a fluid control problem, solve the fluid control problem, and use the solution to obtain a heuristic for the original problem.
We feel that there is an intimate connection between fluid approximations and restless bandits, and that by exploiting it we may be able to investigate asymptotic behavior of such schemes.

8. Brownian Approximations

Brownian or diffusion approximations replace a queueing network, with its jump transitions, by a continuous diffusion process. The approximation agrees with the system on the two first moments and is thus much finer than the fluid approximation (the latter is likened to the law of large numbers, the former to the more informative central limit theorem). Brownian approximations are a cornerstone of applied probability and the literature on them is enormous. Recently Harrison and Wein have used it to great effect in the theory of control of queueing networks.
The scheme is again to replace the queueing network scheduling problem by a Brownian control problem, solve it, and reinterpret the solution as a heuristic for the original problem. Wein[1989] has carried this scheme through and obtained extremely interesting results. Some of these involve priority rules, most are more subtle. The actual implementation of these schemes may be hampered by the excessively large dimensionality of the Brownian control problem. The results so far are however extremely insightful and will no doubt lead to significant progress.

REFERENCES

Adams, J., Balas, E., Zawack, D. (1988) The shifting bottleneck procedure for job shop scheduling. *Management Science* **34**, 391-401.

Baker, K.R. (1974) *Introduction to Sequencing and Scheduling*, Wiley, New York.

Chen, H. (1987) A heterogeneous fluid model and its applications in multiclass queueing networks, preprint.

Chen, H., Mandelbaum, A. (1987) Discrete flow networks: bottleneck analysis and fluid approximations, preprint.

Chen, H. Yao, D.D. (1989) Optimal scheduling control of a multiclass fluid networks, preprint.

Cobham, A. (1954) Priority assignment in waiting line problems, *Operations Research* **2**, 70-76.

Conway, R.W., Maxwell, W.L., Miller, L.W. (1967) *Theory of Scheduling*. Addison Wesley, Reading MA.

Frenk, J.B.G., Rinnooy Kan, A.H.G. (1987) The asymptotic optimality of the LPT rule, *Mathematics Operations Research* **14**, 241-254.

Gershwin, S.B. (1984) A hierarchical framework for discrete event scheduling in manufacturing systems, *IIASA Workshop on Discrete Systems: Models and Applications*, August 3-7, Sopron, Hungary.

Gittins, J.C. (1979) Bandit processes and dynamic allocations indices (with discussion), *J. Royal Statistical Society Series B* **41** 148-177.

Gittins, J.C. (1989) *Bandit Processes and Dynamic Allocation Indices*. Wiley, NewYork.

Greenberg, B.S., Wolff, R.W. (1987) Optimal order of servers for tandem queues in light traffic, *Management Science* **34**, 500-508.

Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimality, *Operations Research* **23**, 270-282.

Harrison, J.M. (1988) Brownian models for queueing networks with heterogeneous customer populations, in W. Fleming, P.L. Lions (eds.)*Stochastic Differential Systems, Stochastic Control Theory and Applications*, IMA Volume 10, Springer, New York, 147-186.

Harrison, J.M., Wein, L.M. (1988) Scheduling of queues: heavy traffic analysis of a simple open network, preprint.

Huang, C.C., Weiss, G. (1989) On the optimal order of M machines in tandem, *Operations Research Letters* to appear.

Kämpke, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines, *J. Applied Probability* **24**, 430-447.

Karmarkar, N., Karp, R. (1982) The differencing method for set partitioning, report UCB/CSD 82/113, computer science division, university of California, Berkeley.

Kawaguchi, T., Kyan, S. (1986) Worst case bound of an LRF schedule for the mean weighted flowtime problem, *SIAM J. Computing* **15**, 1119-1129.
Kelly, F.P. (1979) *Reversibility and Stochastic Networks.* Wiley, New York.
Kleinrock, L. (1976) *Queueing Systems,Volume 2 : Computer Applications.* Wiley, New York.
Klimov, G.P. (1974) Time sharing service systems, *Theory Probability Applications* **19** 532-551.
Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (1989) Sequencing and scheduling algorithms and complexity. Matematisch Centrum report, Amsterdam, Holland.
Leachman (1989)
Matloff, N. (1988) On the value of predictive information in a scheduling problem, *Performance Evaluation* to appear.
Meilijson, I., Weiss, G. (1975) A simple solution to a time sharing problem, Conference paper.
Meilijson, I., Weiss, G. (1977) Multiple feedback at a single server station, *Stochastic Processes Applications* **5**, 195-205.
Muth, E.J. (1979) The reversibility property of production lines, *Management Science* **25**, 152-158.
Nash, P., Gittins, J. (1977) A Hamiltonian approach to optimal stochastic resource allocation, *Advances Applied Probability* **9**, 55-68.
Newell, G.F. (1982) *Applications of Queueing Theory.* Chapman and Hall, London.
Optima
Roundy (1989)
Schrage, E.L., Miller, L.W. (1968) The queue M/G/1 with the shortest remaining processing time discipline, *Operations Research* **16**, 670-685.
Sevcik, K.C. (1974) Scheduling for minimum total loss using service time distributions, *J. Association Computing Machines* **21**, 66-75.
Shantikumar, J.G., Yamazaki, G., Sakasegawa, H. (1989) Characterization of optimal order of servers in a tandem queue with blocking, preprint.
Smith, D.R., Weiss, G. (1989) unpublished work.
Tcha, D.W., Pliska, S.R. (1977) Optimal control of single server queueing networks and multi class M/G/1 queues with feedback, *Operations Research* **25**, 248-258.
Tembe, S.V., Wolff, R.W. (1974) The optimal order of service in tandem queues, *Operations Research* **24**, 824-832.
Varaiya, P.P., Walrand, J., Buyukkoc, c. (1985) Extensions of the multiarmed bandit problem: the discounted case, *IEEE Transactions Automatic Control* **30**, 426-439.
Weber, R.R. (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime, *J. Applied Probability* **19**, 167-182.
Weber, R.R., Varaiya, P., Walrand, J. (1986) Scheduling jobs with stochastically ordered processing times on ;arallel machines to minimize expected flowtime, *J. Applied Probability* **23**, 841-847.
Weber, R.R., Weiss, G. (1989) On an index policy for restless bandits, *J. Applied Probability* to appear.
Wein, L.M. (1988) Ordering tandem queues in heavy traffic, technical report, Sloan school of business, MIT, Cambridge, MA.
Wein, L.M. (1989) Scheduling networks of queues: heavy traffic analysis of a multistation network with controllable inputs. Technical Report, Sloan School of Management, MIT,Cambridge MA.
Weiss, G. (1988) Branching bandit processes, *Probability Engineering Informational Sciences* **2**, 269-278.
Weiss, A., Mitra, D. (1988) A transient analysis of a data network with a processor sharing switch, *Bell Laboratories Technical Report.*
Weiss, G. (1988) Approximation results in parallel machines stochastic scheduling, *Annals Operations Research* to appear.
Weiss, G. (1988) Turnpike optimality of Smith's rule in parallel machines stochastic scheduling, technical report J-88-9, School of ISyE, Georgia Tech, Atlanta GA.
Weiss, G., Pinedo, M.L. (1980) Scheduling tasks with exponential service times on non identical machines to minimize various cost functions, *J. Applied Probability* **17**, 187-202.
Whitt, w. (1985) Best order of queues in series, *Management Science* **31**, 475-487.
Whittle, P. (1980) Multi armed bandits and the Gittins index, *j. Royal Statistical Society, Series B* **42**, 143-149.
Whittle, P. (1988) Restless bandits: activity allocation is a changing world, in *A Celebration of Applied Probability* (ed J. Gani) J. Applied Probability Special Volume **25A**, 287-298.
Yamazaki, G., Sakasegawa, H. (1975) Properties of duality in tandem queueing systems, *Annals Institute Statistical Mathematics* **27**, 201-212.
Yamazaki, G., Sakasegawa, H., Shantikumar, J.G. (1989) on optimal arrangement of stations in tandem queueing systems with blocking, Technical report, University of California, Berkeley.

Probability in the Engineering and Informational Sciences, 3, 1989, 89–116. Printed in the U.S.A.

SCHEDULING STOCHASTIC JOBS WITH A TWO-POINT DISTRIBUTION ON TWO PARALLEL MACHINES

E.G. COFFMAN, JR.
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

M. HOFRI
Computer Science Department
The Technion
Haifa, Israel

G. WEISS
School of ISyE
Georgia Institute of Technology
Atlanta, Georgia 30320

We analyze the optimal preemptive sequencing of n jobs on two machines to minimize expected total flow time. The running times of the jobs are independent samples from the distribution $Pr(X = 1) = p$, $Pr(X = k + 1) = 1 - p$. We verify that the shortest-expected-remaining-processing-time (SERPT) policy, which is optimal for independent and identically distributed (i.i.d.) running times with a monotone hazard-rate distribution, is not optimal for this distribution. However, we prove that if $p \geq 1/k$, then the number of decisions where SERPT and an optimal policy disagree is bounded by a constant independent of n. For $p < 1/k$, we prove that the *expected* number of such decisions has a similar bound. In addition, bounds on the expected increase in flow times under SERPT are derived; these bounds are also independent of n.

1. INTRODUCTION

We consider an instance of a well-known problem in stochastic scheduling theory: $n \geq 1$ jobs are to be run on one or more identical machines operating in parallel. The running times $X_1, \ldots, X_n$ are not known in advance, but they are known to be independent samples from a given distribution $G(x)$. The problem is to find a preemptive scheduling algorithm that minimizes the expected sum of finishing times (total flow time).

The single-machine problem was solved by Sevick [12] for general $G(x)$. Effectively, a ranking of jobs based on elapsed running time is calculated dynamically. At all times, the job assigned to the machine is one having the least rank. As a result of subsequent extensions to other stochastic optimization problems, this rank is now known as a special case of the Gittins index [6].

Weber [13] solved the problem for two or more machines when $G(x)$ has a monotone hazard rate. He showed that the dynamic SERPT (shortest expected remaining processing time) ordering of job assignments is optimal. This algorithm reduces to either non-preemptive sequencing or processor sharing according to whether the hazard rate is increasing or decreasing, respectively. The multiple-machine problem for general $G(x)$ remains open and appears to be very difficult. In particular, no concrete measure of the complexity of this problem is currently known, e.g., formulating the problem as a "game against nature" [10] or finding an imbedded NP-complete problem have so far been unproductive approaches.

The contribution of this paper is an analysis of the two-machine case for a distribution $G(x)$ with a non-monotone hazard rate; we have chosen a distribution simple enough to be tractable, but one that models interesting practical applications. Specifically, $G(x)$ is the two-point distribution $Pr\{X = 1\} = p$ and $Pr\{X = k + 1\} = 1 - p$ with $0 < p < 1$ and $k \geq 2$ an integer. The hazard rate of $G(x)$ is

$$\frac{Pr\{X = j + 1\}}{Pr\{X > j\}} = \begin{cases} p & j = 0 \\ 0 & 1 \leq j < k \\ 1 & j = k \end{cases}$$

which is neither increasing nor decreasing if $k \geq 2$.

We study scheduling policies satisfying the two properties:

(P1) Neither machine is allowed to remain idle while unfinished jobs remain.

(P2) A job can be preempted only at the point when it has received its first unit of service and requires k units more, i.e., at the earliest point when it becomes known whether a job is *short* (requires only one time unit) or *long* (requires $k + 1$ time units).

It is not difficult to prove that (P1) is not restrictive, i.e., an optimal algorithm with this property can always be found. We suspect that (P2) is also not restrictive. However, a proof of this appears difficult.

In spite of the simplicity of the two-point distribution, subsequent sections show that the optimization problem remains nontrivial. Our main results are strong asymptotic characterizations of optimal and SERPT (shortest expected remaining processing time) policies. For $p \geq 1/k$, we prove a *turnpike* theorem; i.e., whenever the number of remaining, unstarted jobs is sufficiently large, SERPT decisions are optimal. For $p < 1/k$, we show that the expected number of decisions where SERPT is not optimal is bounded by a constant. We also bound the expected difference in flow times produced by optimal and SERPT rules. This asymptotic analysis of an approximation algorithm appears to be new in stochastic scheduling theory and to hold promise for the study of near-optimal algorithms for similar problems.

Our assumptions model inspection/service applications in which customers, devices, etc., are first inspected and then serviced, e.g., repaired, if necessary (an event of probability $1 - p$). The constant inspection time is taken as the unit of time. We approximate the service time k as a fixed multiple of the inspection time, so our model applies primarily to situations in which the service time is large compared to the inspection time. Other potential applications arise in computer systems, where the requirements of users break down into short jobs that can be processed very quickly by the operating system, and long jobs that require the operation of a programing system or a program created by the user. Sevcik [12] mentions other computer scheduling applications.

The literature contains a number of results related to our problem within the general theme of parallel machines, given distributions of independent running times, and the objectives of minimizing either the expected sum (possibly weighted) or the maximum of finishing times. For example, see [1,2,4,7,8, 14,15]. However, virtually all of the results to date are characterized by distributions with monotone (possibly weighted) hazard rates or linear orderings by stochastic dominance. As a consequence, the optimal algorithms have all had a simple ranking structure, as in Sevcik's algorithm, where the rank of a job is determined by a relatively easily computed function of job and machine parameters. The ideas underlying these algorithms for the single-machine case were developed by Gittins [6] and were also suggested by the earlier work of Chazan et al. [3] and Meilijson and Weiss [9] on the scheduling of feedback queues. The results in this paper illustrate the substantial increases in the complexity of scheduling parallel machines, when the above simplifying distributional assumptions do not apply.

The remainder of the paper is organized as follows. In the next section, the mathematical model is formalized. In Section 3, a turnpike theorem for the case $p \geq 1/k$ is proved. Section 4 deals with the case $p < 1/k$, and concluding remarks are given in Section 5.

2. DEFINITIONS

We begin by defining the state space and the corresponding stochastic scheduling process. *Decision states* occur only at integer points when one or both machines finish the first or last time unit of a job. They accumulate all of the available running-time information on which to base a scheduling (assignment) decision. For our problem, decision states are defined by triples (c, I, T), where

(i) $0 \le c \le k$ is the number of units of elapsed time already received by the job assigned to the occupied machine, if any. The null value $c = \lambda$ signifies that both machines are available for assignment at a decision point.

(ii) I is a set, possibly empty, of positive integers indexing the unstarted or *new* jobs remaining.

(iii) T is a set, possibly empty, of positive integers indexing the unassigned long jobs having already received their first unit of service (the last k time units of such a job is called a *tail job*). The sets I and T are disjoint.

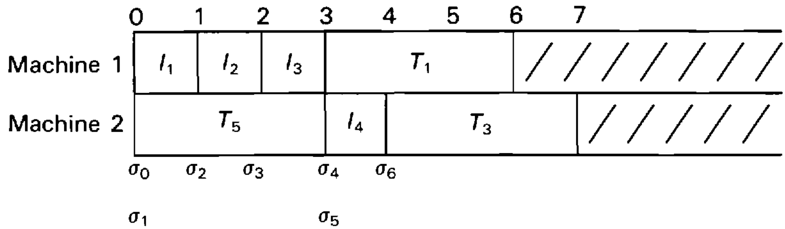
Figure 2.1 illustrates the scheduling process and many of the definitions to follow. Note that I_j denotes the first time unit of X_j, and if X_j is long, then T_j denotes its tail job. The symbol σ is used frequently to denote a decision state.

Observe that if $2 \le c \le k$, then the occupied machine is running a tail job with $k - c + 1$ time units left to run; if $c = 1$, then a tail job is just being started on the occupied machine; and if $c = 0$ the first time unit of a new job is just being started on the occupied machine. Thus, decision states in which $c = 0$ or 1 are immediately preceded by decision states in which $c = \lambda$.

The "standard" initial state is $(\lambda, \{1,2,\ldots,n\}, \phi)$ where both machines are available and there are no partially run (i.e., tail) jobs. However, to simplify inductive arguments, it is convenient to allow more general initial states. In Section 3, we allow any state (c, I, T), where $|I| + |T| > 0$ and $c = \lambda$ or $1 \le c \le k$; i.e., either both machines are available or one of them is unavailable for the first $k - c + 1$ time units (as if it were finishing some artificial tail job). In Section 4, we also allow $c = 0$, in which case the initial delay on the occupied machine is a random variable: it is 1 with probability p and $k + 1$ with probability $1 - p$. Thus, $c \ne \lambda$ in the initial state may be interpreted as the elapsed time of some artificial new job which is running non-preemptively.

Clearly, the states of the form (c, ϕ, T), $|T| \ge 1$, and (c, I, ϕ), $|I| \ge 1$, allow but one decision: assign a tail or new job, respectively. Final decisions take place in states of the form (c, I, ϕ), $|I| = 1$, or (c, ϕ, T), $|T| = 1$.

In the sample schedules under some policy A, let $F_1, F_2, \ldots, F_{n+r}$ denote the finishing times of the n new jobs and the r tail jobs specified in the initial state $\sigma = (c, I, T)$, $|I| = n$, $|T| = r$. The random variable $A(\sigma) = \sum_{i=1}^{n+r} F_i$ de-



Sample values X_1 X_2 X_3 X_4 (Probability $= p^2(1-p)^2$)
4 1 4 1

T_5 = tail job in initial state
I_j = initial unit of X_j
T_j = tail of $X_j = k = 3$.

Decision states: $\sigma_0 = (\lambda, \{1,2,3,4\}, \{5\})$, $\sigma_1 = (1, \{1,2,3,4\}, \phi)$,
$\sigma_2 = (2, \{2,3,4\}, \{1\})$, $\sigma_3 = (3, \{3,4\}, \{1\})$,
$\sigma_4 = (\lambda, \{4\}, \{1,3\})$, $\sigma_5 = (1, \{4\}, \{3\})$, $\sigma_6 = (2, \phi, \{3\})$.

Flow time $= 6 + 2 + 7 + 4 + 3 = 22$

Figure 2.1. A sample function for $k=3$ and initial state $(\lambda, \{1,2,3,4\}, \{5\})$.

notes the *total flow time* (or simply *flow time*) under A. OPT refers to a policy that minimizes $EA(\sigma)$, where the expectation is over all $X_1, \ldots, X_n$.

We let f^A denote the decision rule of policy A; i.e., A's decision in state $\sigma = (c, I, T)$ is denoted by $f^A(\sigma)$, which is an integer in I or T. If $i = f^A(\sigma) \in I$, then A makes an *I-decision* and assigns I_i to an available machine; if $i \in T$, then A makes a *T-decision* and assigns T_i to an available machine. (The choice between two available machines can be arbitrary, since it obviously has no effect on flow times.)

Clearly, the order in which new jobs are selected for assignment is unimportant; such jobs are stochastically identical by definition. It is also clear that the expected total flow time starting in state (c, I, T) depends only on c, $|I|$, $|T|$. These observations are exploited in Section 4, where a simpler decision state is adopted. In particular, only the *numbers* of new jobs and tail jobs are combined with the elapsed-time parameter c in a decision state. The more detailed state (c, I, T) is used in Section 3, because it greatly simplifies the proof of the turnpike property for $p \geq 1/k$.

SERPT is an important policy to consider for our problem, because it is optimal when $G(x)$ has a monotone hazard rate, and it is optimal on a single machine when $G(x)$ is our two-point distribution. Specialized to our setup, SERPT reduces either to non-preemptive sequencing (NS) or to preemptive sequencing (PS). According to NS, once a job is assigned, it is run to completion without interruption. In all states (c,I,T), $|I| \geq 1$, policy PS always preempts a long job after its first time unit and assigns a new job in its place.

NS applies under SERPT when $k < EX$ (i.e., when the duration of a tail job is less than the expected total time of a new job), and PS applies when $EX \leq k$. Since $EX = 1 + (1-p)k$, we see that SERPT reduces to NS when $p < 1/k$, and to PS when $p \geq 1/k$.

It is easily verified that SERPT does not minimize expected flow time in general. Indeed, for any $0 < p < 1/2$ and $k > 1$, there are states in which SERPT decisions are not optimal.

Example 1: Consider the initial state $\sigma = (\lambda,\{1\},\{2,3\})$. If $p < 1/k$, then SERPT reduces to NS and the sample schedules are those in Figure 2.2(a). OPT is shown in Figure 2.2(b) and yields $E[\text{NS}(\sigma) - \text{OPT}(\sigma)] = p(k-1)$. Note that state σ is reached from the standard initial state $(\lambda,\{1,2,3\},\phi)$ when the first two jobs scheduled are X_2 and X_3, and both are long.

Example 2: Consider the initial state $\sigma = (\lambda,\{1,2\},\{3\})$ which can be reached from $(\lambda,\{1,2,3,4\},\phi)$ if I_3 and I_4 are assigned first and one of X_3 and X_4 is long and the other is short. If $p \geq 1/k$, then SERPT reduces to PS, and if in addition $p < 1/2$, then we find $E[\text{PS}(\sigma) - \text{OPT}(\sigma)] = (1-p)(1-2p) > 0$, as shown in Figure 2.3.

There are many other examples showing that SERPT is not optimal. For example, the initial state in Example 1 can be generalized to $\sigma' = (\lambda,\{1\},T)$ for $|T| \geq 2$ even. For p sufficiently near but still less than $1/k$, we again obtain $E[\text{NS}(\sigma') - \text{OPT}(\sigma')] > 0$. Other examples will be given in the context of the analysis in Section 3.

3. THE OPT AND PS POLICIES FOR $p > 1/k$

Recall that in this section, if (c,I,T) is an initial state, then $c \neq 0$. In the inductive proofs to follow, the size of a state (c,I,T) refers to its position in the lexicographic ordering of $(|I|,|T|)$.

The asymptotic results in this section are considerably more precise than those for NS in Section 4. In particular, we show that, if $p > 1/k$, then all sample schedules under an optimal policy must have the general form illustrated in Figure 3.1. That is, a sample schedule must begin with at least $n-k$ I-decisions in an initial region R_1, end with only T-decisions in a final region R_3, and contain a region R_2 between R_1 and R_3, in which the remaining I-decisions and at most k T-decisions are made. Thus, since PS decisions differ from those of OPT only in the at most k T-decisions in R_2, PS is an optimal turnpike policy.

Example 3: It is tempting, perhaps, to conjecture that the region R_2 always consists of at most one T-decision. A smallest counterexample is provided by the initial state $\sigma = (\lambda,\{3,4,5\},\{1\})$, which is reached from the standard initial

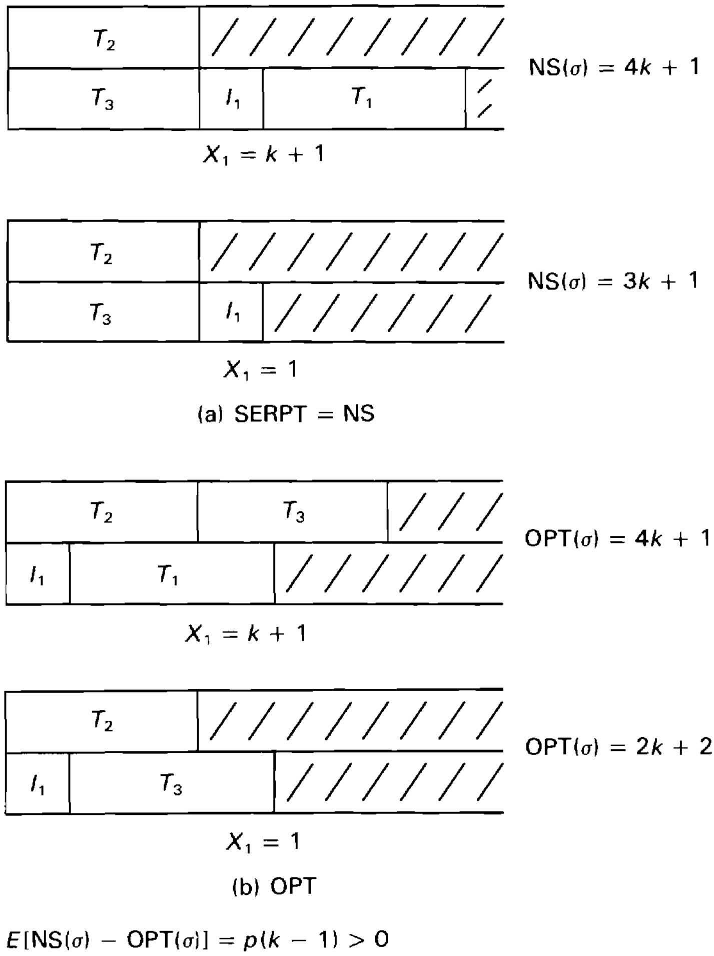


FIGURE 2.2. Counterexample for $p < 1/k$; $\sigma = (\lambda, \{1\}, \{2,3\})$.

state $(\lambda, \{1,2,3,4,5\}, \phi)$ when I_1 and I_2 are run first and X_1 and X_2 are long and short jobs, respectively. An optimal policy can be worked out in the usual way by Bellman equations; a general solution for any initial state can be written, but the expressions are awkward and uninstructive and therefore omitted. A complete analysis shows that for all $1/k < p < 1/(k-1)$, $k \geq 7$, OPT begins with a T-decision on one machine and two consecutive I-decisions on the other. If both of the I-decisions reveal long jobs (thus reaching a state $\sigma' = (3, I', T')$ with $|I'| = 1$ and $|T'| = 2$), OPT again makes a T-decision instead of the one available I-decision. That is, twice, when in a state with an I-decision available (namely, states σ and σ'), OPT chooses a T-decision.

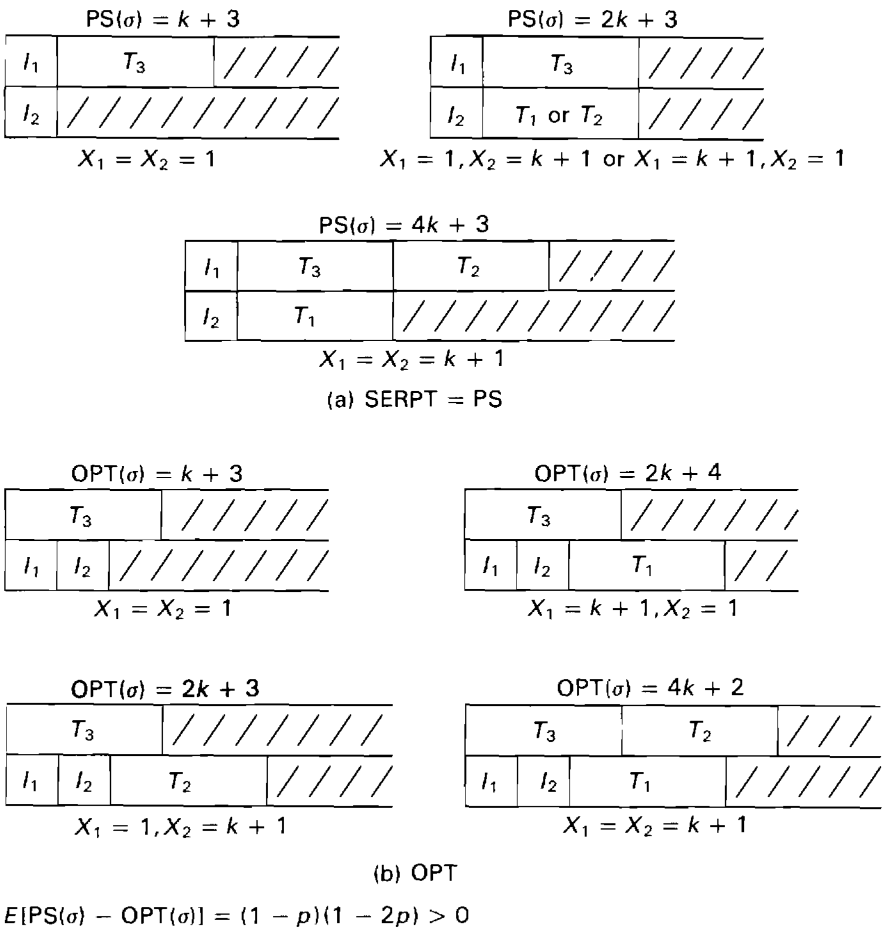


FIGURE 2.3. Counterexample for $1/k \le p < 1/2$; $\sigma = (\lambda, \{1,2\}, \{3\})$.

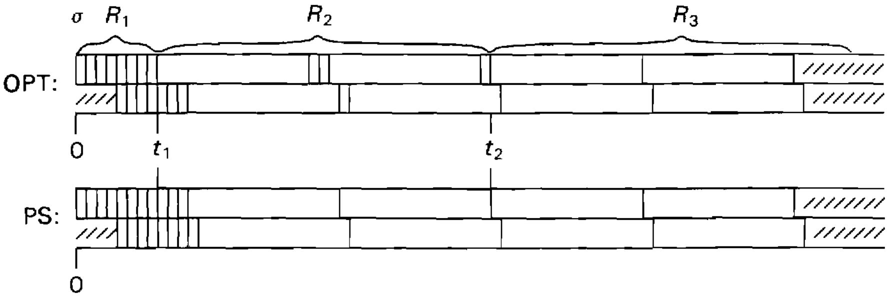


FIGURE 3.1. Sample schedules under OPT and PS.

To prove the properties implicit in Figure 3.1, we begin with two preliminary results having similar, essentially combinatorial proofs. Our first lemma shows that if $p > 1/k$, then OPT never assigns two tail jobs that begin at the same time, unless there is no new job waiting.

LEMMA 3.1: *If $p > 1/k$, then an I-decision is made by OPT in any state $(1, I, T)$ with $|I| > 0$.*

PROOF: The result is obvious if $T = \phi$, so let $|T| \geq 1$ and let $(1, I, T)$ be the smallest state violating the lemma. Without loss of generality, let $\sigma_0 = (1, I, T)$ be the initial state at $t = 0$. At time k, OPT must reach the state $\sigma_1 = (\lambda, I, T - \{j\})$, where $j = f(\sigma_0)$. Since σ_0 is the smallest counterexample, at least one I-decision must be made at time k. Thus, if σ_2 denotes the state following σ_1 at time k, we can assume $i = f(\sigma_2) \in I$, since the ordering of the two assignments at time k can be arbitrary. At the same time, $f(\sigma_1) = l$ with l belonging to $I - \{i\}$ or to $T - \{j\}$.

We now construct a policy A that reverses the order of T_j and I_i, but otherwise preserves the assignments of OPT. Figure 3.2 illustrates the construction.

For A's decision at time 0 in state σ_0, we define the I-decision

$$f^A(1, I, T) = i. \tag{3.1}$$

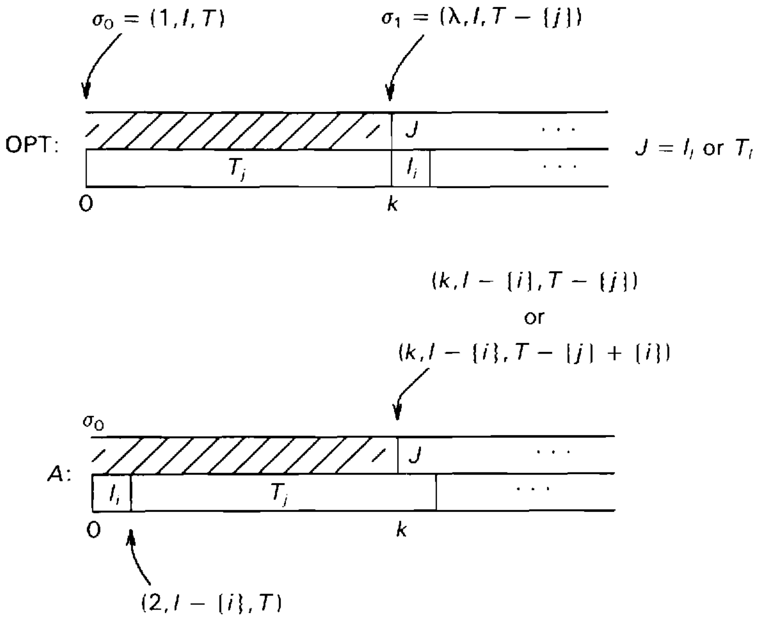


FIGURE 3.2. Comparison for Lemma 3.1.

For A's decision at time 1, we want a T-decision irrespective of whether X_i is short or long; so we define

$$f^A(2, I - \{i\}, T) = f^A(2, I - \{i\}, T + \{i\}) = j. \tag{3.2}$$

Apart from the above interchange of I and T decisions, we want A to simulate OPT. For this purpose, we first set up the appropriate decision at time k, which again must be independent of whether X_i is short or long. We define

$$f^A(k, I - \{i\}, T - \{j\}) = f^A(k, I - \{i\}, T - \{j\} + \{i\}) = l, \tag{3.3}$$

where $l = f(\lambda, I, T - \{j\})$ (either $l \in I$ or $l \in T$). Finally, for states σ reachable from σ_0, other than those in Eqs. (3.1)–(3.3), we define

$$f^A(\sigma) = f(\sigma). \tag{3.4}$$

It is clear from Eqs. (3.1)–(3.3) that in $[0, k+1]$ both A and OPT run I_i, T_j, and one time unit of X_l (either I_l, if $l \in I$, or the first time unit of T_l, if $l \in T$). Thus, the decision state reached by A at the completion of T_j is the same as the decision state reached by OPT at the completion of I_i. By Eqs. (3.1)–(3.4), the only flow times that can differ under A and OPT are those of X_i and X_j. A delays the flow time of X_j by one time unit relative to OPT. But if X_i is short, then OPT delays the flow time of X_i by k time units relative to A. If X_i is long, it has the same flow time under A and OPT. Then

$$E[A(\sigma_0) - \mathrm{OPT}(\sigma_0)] = 1 - pk < 0, \tag{3.5}$$

which contradicts the optimality of OPT. ■

The next result shows that if $p > 1/k$, then OPT cannot make two consecutive T-decisions if the first occurs in a state (c, n, r) with $2 \le c \le k$ and $n \ge 1$. In the remainder of the paper, it is convenient to define

$$c + 1 = \lambda \quad \text{if } c = k, \qquad c - 1 = k \quad \text{if } c = \lambda. \tag{3.6}$$

LEMMA 3.2: *If $p > 1/k$ and OPT makes a T-decision in some state (c, I, T) with $2 \le c \le k$, $|I| \ge 1$ and $|T| \ge 1$, then OPT must make an I-decision in the next state, $k - c + 1$ time units later.*

PROOF: The result is trivial if $|T| = 1$, so let $|T| \ge 2$ and let (c, I, T) be the smallest state violating the lemma. Again, we may suppose without loss of generality that $\sigma_0 = (c, I, T)$ is the initial state. Let $f(\sigma_0) = j$, so that by assumption, in the next state $\sigma_1 = (k - c + 2, I, T - \{j\})$, we have $l = f(\sigma_1) \in T$.

Since σ_0 is the smallest counterexample, OPT must make an I-decision at time k in state $\sigma_2 = (c, I, T - \{j, l\})$. Let $i = f(\sigma_2) \in I$. As in Lemma 3.1, we construct a policy A that reverses the order of a T and I-decision. Figure 3.3 illustrates the construction.

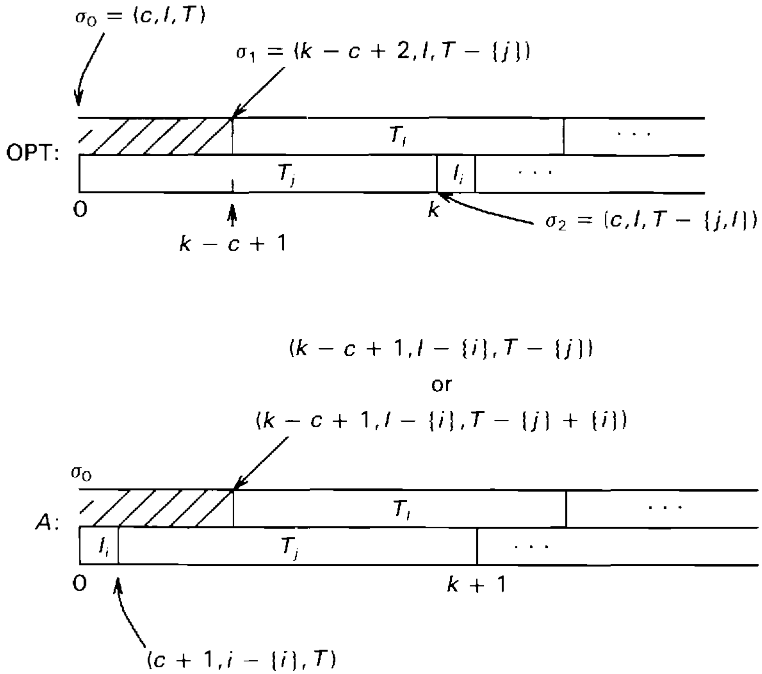


Figure 3.3. Comparison for Lemma 3.2.

We define

$$f^A(c, I, T) = i, \tag{3.7}$$

$$f^A(c+1, I-\{i\}, T) = f^A(c+1, I-\{i\}, T+\{i\}) = j, \tag{3.8}$$

$$f^A(k-c+1, I-\{i\}, T-\{j\}) = f^A(k-c+1, I-\{i\}, T-\{j\}+\{i\}) = l, \tag{3.9}$$

and for states reachable from σ_0, other than those in Eqs. (3.7)-(3.9), we define

$$f^A(\sigma) = f(\sigma). \tag{3.10}$$

In analogy with Lemma 3.1, Eqs. (3.7) and (3.8) implement the desired interchange of I- and T-decisions, while Eqs. (3.9) and (3.10) assure that A otherwise simulates OPT. It is easy to see that both A and OPT reach the same state at time $k+1$. As in Lemma 3.1, the interchange of I- and T-decisions again produces the contradiction in Eq. (3.5). ■

Our next result shows that if OPT assigns a tail job at some time t on one of the machines, then the time available to run waiting jobs in the interval $[t, t+k]$ on the other machine cannot be less than the number of new jobs remaining at time t; i.e., it must be possible during $[t, t+k]$ to run the initial time units of all new jobs waiting at time t. A weak turnpike property of PS follows immediately from this result: In any state (c, I, T) with $|I| > k$, an I-decision is optimal. After combining this result with Theorem 3.2, we will obtain the stronger property. The number of times OPT makes a non-PS decision (i.e., does not preempt when at least one new job is waiting) is bounded independent of the initial state.

THEOREM 3.1: *If $p \geq 1/k$ and if OPT makes a T-decision in state (c, I, T), then $|I| \leq c-1$ if $c \neq 0$ (by Eq. (3.6), $c-1=k$ if $c=\lambda$), and $|I| \leq k-1$ if $c=0$.*

PROOF: Let (c, I, T) be the smallest state violating the theorem. If $c=0$, then (c, I, T) must be preceded immediately by a state $(\lambda, I+\{l\}, T)$ for some l. In this case, we begin by modifying OPT so that it reverses the decisions in these two states; i.e., $f(\lambda, I+\{l\}, T) = j \in T$ and $f(1, I+\{l\}, T-\{j\}) = l$. Clearly, the flow times under OPT are unaffected and the violation of the theorem now occurs in the state $(\lambda, I+\{l\}, T)(|I| > k-1$ becomes $|I+\{l\}| > k)$. It follows that we may assume an algorithm OPT such that $\sigma_0 = (c, I, T)$, $c \neq 0$, is the smallest *initial* state violating the theorem; i.e., $|I| > c-1$ and $|T| \geq 1$.

Since we are assuming $|I| > c-1$ and a T-decision at time 0, there must be at least one I-decision at time k in every sample schedule, for otherwise, Lemma 3.1 or 3.2 would be violated in a state reachable by OPT at time k. Since the order in which new jobs are assigned cannot affect expected total flow times, we may choose I_i as an assignment made by OPT at time k in *every* sample schedule starting in state σ_0. For the case (if it can occur) when OPT makes two assignments at time k, I_i is taken to be the second assignment.

Let $j = f(\sigma_0)$ be OPT's T-decision at time 0. As in Lemmas 3.1 and 3.2, we construct a policy A that simulates OPT except for an interchange of I- and T-decisions. We consider the cases $c \neq \lambda$ and $c = \lambda$ separately. Figure 3.4 illustrates the construction.

Case 1 ($c \neq \lambda$): For A's decisions at times 0 and 1, we define

$$f^A(c, I, T) = i, \tag{3.11}$$

$$f^A(c+1, I-\{i\}, T) = f^A(c+1, I-\{i\}, T+\{i\}) = j, \tag{3.12}$$

thus implementing the desired interchange, independently of whether X_i is short or long. Now with one exception, let $\sigma' = (c', I', T')$ be any state reachable by OPT at some time $t \in \{k-c+1, \ldots, k\}$. The exception occurs when $c' = \lambda$, and hence $t = k$; in this case, σ' can only be the first decision state at time k. (Recall that I_i is assigned by OPT in the second state at time k, if both machines become available at time k.)

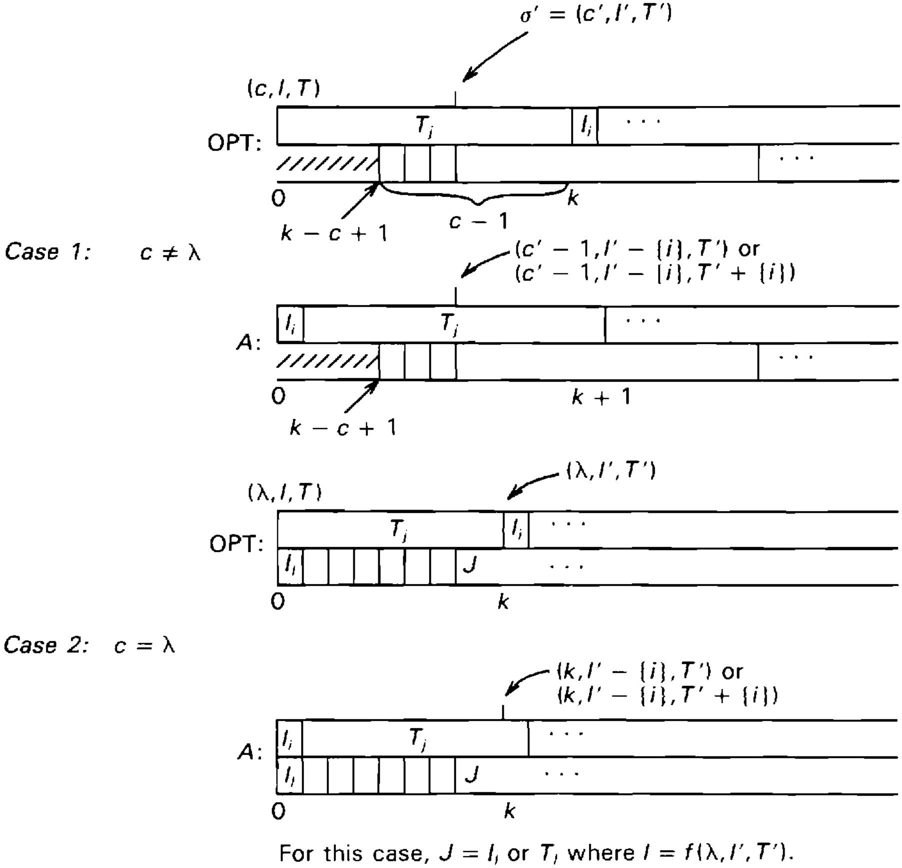


FIGURE 3.4. Pairs of schedules for A and OPT under the contradiction of Theorem 3.1.

Note that OPT must reach σ' from σ_0 by $|I| - |I'|$ consecutive I-decisions, where $|I| - |I'| = t - (k - c + 1) \leq c - 1$. We define for each such state σ'

$$f^A(c' - 1, I' - \{i\}, T') = f^A(c' - 1, I' - \{i\}, T' + \{i\}) = f(\sigma'). \quad \textbf{(3.13)}$$

This definition reflects the fact that if A and OPT are to make the same decision at time t on the machine not running T_j, then the state under A must indicate that one fewer unit of time has elapsed for T_j, that I_i has already been run, and that there is an additional tail job (T_i) if running I_i revealed X_i as a long job. Note that if $|I| - |I'| = c - 1$ (i.e., $t = k$), then $c' = \lambda$ and $c' - 1 = k$;

in this case, $f(\lambda, I', T') \neq i$, since by assumption I_i is the second assignment at time k.

In states σ reachable from σ_0, other than those in Eqs. (3.11)–(3.13), we define

$$f^A(\sigma) = f(\sigma). \tag{3.14}$$

It is clear that, apart from the interchange in Eqs. (3.11) and (3.12), A simulates OPT. In particular, the decision state reached by A at the completion of T_j is the same as the decision state reached by OPT at the completion of I_i, assuming the same sample of new jobs indexed by I. As in Lemmas 3.1 and 3.2, we obtain the contradiction in Eq. (3.5).

Case 2 ($c = \lambda$): By Lemma 3.1, OPT's two assignments at time 0 are T_j and some new job, in that order. Let $l = f(1, I, T - \{j\}) \in I$ index the new job assigned at time 0. For A's decisions at time 0, we define

$$f^A(\lambda, I, T) = i, \qquad f^A(0, I - \{i\}, T) = l.$$

For A's first decision at time 1, we define for each $Z = \phi, \{i\}, \{l\}, \{i,l\}$, the T-decision

$$f^A(\lambda, I - \{i,l\}, T + Z) = j.$$

For A's second decision at time 1, we define

$$\begin{aligned} f^A(1, I - \{i,l\}, T - \{j\}) &= f^A(1, I - \{i,l\}, T - \{j\} + \{i\}), \\ &= f(2, I - \{l\}, T - \{j\}), \end{aligned}$$

for the case when X_l is short, and

$$\begin{aligned} f^A(1, I - \{i,l\}, T - \{j\} + \{l\}) &= f^A(1, I - \{i,l\}, T - \{j\} + \{i,l\}), \\ &= f(2, I - \{l\}, T - \{j\} + \{l\}), \end{aligned}$$

for the case when X_l is long. In both cases, the decision may be an I-decision or a T-decision. By inspection, it can be seen that A makes all of the assignments made by OPT at times 0 and 1, for any given sample of new jobs. Policy A also assigns I_i but delays the start of T_j by one time unit. The remainder of the construction follows that of case 1. Again, A is uniquely defined and A and OPT converge to the same state after A completes T_j and OPT completes I_i. We again obtain the contradiction in Eq. (3.5). ∎

The next theorem further restricts the structure of OPT schedules when $p > 1/k$. With this added restriction a strong turnpike property can be proved, and a simple bound can be derived on the expected increase in total flow time incurred by PS.

THEOREM 3.2: *If $p > 1/k$, then in any optimal decision sequence starting in state σ, there are at most k T-decisions while at least one new job is waiting. In addition,*

$$E[\text{PS}(\sigma) - \text{OPT}(\sigma)] \le k(k+1)/2. \tag{3.15}$$

PROOF: By Theorem 3.1, an OPT schedule must begin with a region R_1 containing at least $n - k$ new-job assignments before the first tail job is started, at t_1 say. A region R_2 then extends from t_1 to the time t_2, when the last new job completes its initial time unit (see Figure 3.1). We now show that at most k tail jobs are started in R_2.

Let $\tau_1, \tau_2, \ldots, \tau_l$ be the starting times of tail jobs in R_2 ($\tau_1 = t_1, \tau_l < t_2$), in the order they are assigned. We assume $l \ge 2$, since the first part of the theorem follows trivially from $k \ge 2$ otherwise. We claim that

$$\tau_{j-1} < \tau_j < \tau_{j-1} + k, \qquad 2 \le j \le l. \tag{3.16}$$

Lemma 3.1 establishes the first inequality $\tau_{j-1} < \tau_j$. For the second inequality, suppose $\tau_j \ge \tau_{j-1} + k$. Then in the state (c, I, T) where the $(j-1)$st tail job is assigned, the inequality $|I| \ge c - 1$ must hold (see Figure 3.5). But if $|I| = c - 1$, then τ_{j-1} must be the time of the last T-decision in R_2, and if $|I| > c - 1$, then Theorem 3.1 is violated. Thus, Eq. (3.16) holds.

Now if $l > 2$, then Eq. (3.16) along with Lemma 3.2 implies that at least one I-decision must be made between the $(j-1)$st and jth T-decisions. Thus, on each machine in R_2, T-decisions alternate with sequences of one or more consecutive I-decisions. Since at most k new jobs start in R_2 (by Theorem 3.1 applied to the first T-decision), there can be at most k tail jobs started in R_2. This proves the first part of the theorem.

For the bound given by Eq. (3.15), we observe that the expected flow time of short jobs according to PS is at most that according to OPT. Thus, we can obtain a crude bound by calculating a deterministic worst-case increase in the finishing times of long jobs.

As illustrated in Figure 3.1, consider a sample schedule under OPT and the corresponding schedule under PS. It is easy to verify that we can transform the OPT schedule in R_2 to the corresponding PS schedule in $[t_1, t_2]$ so that at

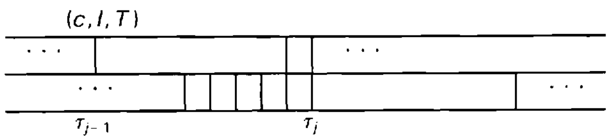


FIGURE 3.5. Example for Theorem 3.2.

most $\left\lceil \frac{l}{2} \right\rceil$ is added to the finishing time of each tail job in R_2. Since $l \leq k$, we have an increase of at most $k^2/2$ in the flow times of tail jobs started in R_2.

Let $m \geq 1$ tail jobs be started in R_3 according to OPT. The starting time of the first tail job started in R_3 is t_2: let t_2' be the starting time of the second, if any. The above transformation from OPT to PS in general changes these starting times to t_3 and t_3', where $t_3 + t_3' = t_2 + t_2'$ (one starting time is shifted left and the other shifted right by an equal amount) and $t_3' - t_3 < k$. This creates no change in the total flow time of tail jobs starting in R_3 if m is even, and a change of at most $k/2$ if m is odd. The increase in the total flow time of long jobs is therefore no greater than $k^2/2 + k/2 = k(k+1)/2$. ∎

In this section, our results have been based on the assumption that $p > 1/k$. This has been for convenience, since we could have also allowed equality, i.e., $p \geq 1/k$. Lemmas 3.1 and 3.2 and Theorems 3.1 and 3.2 would then have been modified to state that *there exists* an optimal policy, rather than *there must be* an optimal policy with the claimed properties. With these changes, we can conclude that PS is a (strongly) optimal turnpike policy for all $p \geq 1/k$.

4. THE OPT AND NS POLICIES FOR $p < 1/k$

The asymptotics in this section are obtained from direct calculations of expected NS flow times. Accordingly, we begin with derivations of the appropriate formulas and a study of their properties.

Given an initial state $\sigma = (c, I, T)$, the expected total flow time is a function only of c, $n = |I|$, and $r = |T|$. In this section, we exploit this fact and use (c,n,r) as the decision state. Policy A is now defined by a simpler mapping f^A, where $f^A(c,n,r)$ takes on one of two values, one calling for a new-job assignment and the other for a tail-job assignment. The terminology of I- and T-decisions is used as before. For the purposes of this section, any order of selection from I and T may be assumed. Initial states (c,n,r) are general with $n \geq 0$, $r \geq 0$ $(n + r > 0)$, and $c = \lambda$ or $0 \leq c \leq k$. We call (n,r) the *backlog* of state (c,n,r). The *level* of (c,n,r) is simply n, the number of new jobs.

Observe that states in which NS assigns a new job must have the form $(c,n,0)$, i.e., no tail jobs are waiting to be assigned. Also, after the first new job has been assigned, the only states reachable by NS have the form $(c,n,0)$, $(c,n,1)$, or $(\lambda,n,2)$.

Under NS, let C_i, $i \geq 0$, denote the elapsed-time random variable in the state where the $(i+1)$st new job is assigned. C_0 corresponds to the initial state if there are no waiting tail jobs in the initial state. By the above observations it is easily verified that, for any given C_0, $\{C_i; 0 \leq i \leq n-1\}$ is a finite, irreducible Markov chain. However, the analysis of NS is more natural in terms of the *remaining* times on the occupied machine, when new jobs are assigned.

$k - C_i + 1$ if $C_i \neq 0$, and $D_i = 1$ or $k + 1$ with probabilities p and $1 - p$, respectively, if $C_i = 0$. After calculating the *conditional* expected flow times

$$E\text{NS}(n,r|d) \equiv E\text{NS}(n,r|D_0 = d),$$

we then obtain

$$\begin{aligned} E\text{NS}(c,n,r) &= E\text{NS}(n,r|k - c + 1), \qquad c \neq 0, \\ E\text{NS}(0,n,r) &= pE\text{NS}(n,r|1) + (1-p)E\text{NS}(n,r|k+1). \end{aligned} \tag{4.1}$$

Before giving these calculations, we discuss briefly the properties of the chain $\{D_i\}$. With D_0 given, the chain is defined by the recurrence

$$D_{i+1} = |D_i - X_{i+1}|, \qquad i = 0,1,2,\ldots, \tag{4.2}$$

and is therefore a finite, irreducible Markov chain on the set $\{0,1,\ldots,k+1\}$. Note that if k is odd, the chain is ergodic. If k is even, then the X_i's must all be odd (each is 1 or $k + 1$), and hence they create an alternating sequence of parities in the D_i's. Thus, the chain has period 2. A brief analysis of the limiting distributions is given in the Appendix. This analysis shows that if

$$\alpha = \frac{1}{2}\left[\frac{1}{\mu} + \mu p(1-p)k^2\right] = \frac{EX^2}{2EX}, \tag{4.3}$$

where $1/\mu \equiv EX$, then as $i \to \infty$,

$$ED_i \to \alpha, \qquad k \text{ odd},$$

$$ED_i + \xi_{i+d}\frac{\mu}{2} \to \alpha, \qquad k \text{ even},$$

where ξ_j is the parity function, $\xi_j = +1$, j even, and $\xi_j = -1$, j odd. As expected, α is simply the mean forward-recurrence time of a renewal process with intervals between renewals having the distribution of job running times.

In calculating expected flow times, we condition on an initial resume delay d on the occupied machine and consider first an initial backlog of $(n,0)$. It is easy to see that the ith job, $i \geq 1$, begins at time $\frac{1}{2}\left[d + \sum_{1\le j\le i-1} X_j\right] - D_{i-1}/2$ with the expected value $\frac{1}{2}[d + (i-1)/\mu] - ED_{i-1}/2$. We add $1/\mu$ to the latter expression to obtain the expected finishing time of the ith job. Thus, for the expected total flow time, we have

$$\begin{aligned} E\text{NS}(n,0|d) &\equiv E\text{NS}(n|d) \\ &= \sum \left[\frac{d + (i-1)/\mu - ED_{i-1}}{2} + \frac{1}{\ }\right]. \end{aligned} \tag{4.4}$$

In the summation of Eq. (4.4), it is convenient to add and subtract $n\alpha/2$ and put the result in the form

$$ENS(n|d) = \frac{n(n+3)}{4\mu} + \frac{n}{2}(d-\alpha) - \frac{1}{2}h_n(d), \tag{4.5}$$

where

$$h_n(d) = \sum_{0 \le i \le n-1} E(D_i - \alpha | D_0 = d). \tag{4.6}$$

Before analyzing the function $h_n(d)$, we extend Eq. (4.5) to initial states with tail-job backlogs $r \ge 1$. By the NS rule these tail jobs are run first. At the point when the tail-job backlog first reduces to 0, the initial resume delay D_0 begins. Clearly, D_0 is a deterministic function of the initial delay d. For $0 \le d \le k$, $D_0 = d$ if $r \ge 1$ is even, and $D_0 = k - d$ if r is odd. If $d = k + 1$, then $D_0 = k - 1$ if $r \ge 1$ is even, and $D_0 = 1$ if r is odd. We denote this relation by $D_0 = \delta(d,r)$, and define $\delta(d,0) = d$. The dependence on d and r will often be suppressed when obvious in context.

A calculation now shows that for $r \ge 1$ and $0 \le d \le k$,

$$\begin{aligned} ENS(n,r|d) = {} & \frac{r}{4}[2d + (r+2)k] + \frac{1}{4}(\delta - d) \\ & + ENS(n|\delta) + \frac{n}{2}[rk - (\delta - d)], \end{aligned} \tag{4.7}$$

where the first two terms on the right-hand side give the expected flow time of the first r tail jobs, the third term is the expected flow time of the n new jobs, assuming that the first new-job assignment is made at time 0, and the last term corrects the preceding term by adding n times the instant, $[rk - (\delta - d)]/2$, when the first new job is actually assigned. An expression for $d = k + 1$, $r \ge 1$ can be found from Eq. (4.7) and

$$ENS(n,r|k+1) = (n+r)k + ENS(n,r-1|1). \tag{4.8}$$

We return now to an analysis of the asymptotics of the expected value $h_n(d)$ in Eq. (4.6). Define

$$h(d) = \frac{\mu}{2}\left(d^2 - \frac{1}{6}\right) - \frac{\mu^2}{6}EX^3, \qquad 0 \le d \le k+1, \tag{4.9}$$

where $EX^3 = p + (1-p)(k+1)^3$. In the Appendix, we prove the following result.

THEOREM 4.1: *If k is odd, then $h_n(d) \to h(d)$ as $n \to \infty$. If k is even, then $h_{2n}(d) \to h(d) + \xi_d\frac{\mu}{4}$ and $h_{2n+1}(d) \to h(d) - \xi_d\frac{\mu}{4}$ as $n \to \infty$.*

The maximum of Eq. (4.9) is obtained at $d = k + 1$. For $p < 1/k$, a calculation yields the bound

$$\max_{0 \le d \le k+1} |h(d)| \le \frac{k+2}{2}. \tag{4.10}$$

Next, we prove two lemmas from which the asymptotic optimality of NS will follow easily. Lemma 4.1 eases the calculations in the proof of Lemma 4.2; this will be further explained just prior to the statement of Lemma 4.2.

For the remaining results of this section, we define $D(\sigma)$, the *final delay* produced by NS starting in state σ, as the delay on the occupied machine until the end of the schedule, measured from the time when the backlog first reaches (0,0); in other words, $D(\sigma)$ is simply the difference in machine finishing times. Note that final delays are distinguished from resume delays by the absence of a subscript. As before, $D(n,r|d)$ and $D(n|d) \equiv D(n,0|d)$ are conditioned on an initial delay of d time units on the occupied machine.

LEMMA 4.1: *Let $h(D(n|d))$ be the random variable given by Eq. (4.9) with d replaced by $D(n|d)$. We have*

$$h_n(d) + Eh(D(n|d)) = h(d). \tag{4.11}$$

PROOF: We write for any $s \ge 1$

$$h_{n+s}(d) = E\left[\sum_{j=0}^{n+s-1} (D_j - \alpha)|D_0 = d\right],$$

$$= h_n(d) + E\left[\sum_{j=n}^{n+s-1} (D_j - \alpha)|D_n(n+s|d)\right].$$

But the second term on the right of this last expression is simply $Eh_s(D_n(n+s|d))$. Since $D_n(n+s|d)$ and $D(n|d)$ are equal in distribution for all $s \ge 1$, we have

$$h_{n+s}(d) = h_n(d) + Eh_s(D(n|d)).$$

Taking the limit $s \to \infty$ through the even integers and applying Theorem 4.1 now yields Eq. (4.11). ■

The notation $D_n^A(\sigma)$ and $D^A(\sigma)$ extends our definitions of resume and final delays to an arbitrary policy A. For any policy A starting in state σ, we define the *modified flow time*

$$A^*(\sigma) = A(\sigma) - \frac{1}{2} h(D^A(\sigma)), \tag{4.12}$$

By Eq. (4.10), the expected difference in $A(\sigma)$ and $A^*(\sigma)$ is bounded by

$$|EA^*(\sigma) - EA(\sigma)| < (k+2)/4. \tag{4.13}$$

The usefulness of the new cost function stems from Eq. (4.13) and the fact, implied by Lemma 4.2 below, that its expected value is minimized by NS.

From Eq. (4.11), we note that $Eh(D^{\text{NS}}(n,r|d)) \to 0$ as $n \to \infty$. Thus, $E\text{NS}^*(n,r|d) \to E\text{NS}(n,r|d)$ as $n \to \infty$ for any fixed r and d. The expected value of the flow time modification and the relation of Eq. (4.11) also have important consequences in the calculation of $E\text{NS}^*(n,r|d)$, which we now describe.

Observe that $E\text{NS}(n|d)$ fails to be an explicit function of n because of the term $-\frac{1}{2}h_n(d)$ in Eq. (4.5). However, if we further subtract $\frac{1}{2}Eh(D(n|d))$, then the two terms combine into $-\frac{1}{2}[h_n(d) + Eh(D(n|d))]$, which by Eq. (4.11) is an explicit function of n. Since the subtraction of $\frac{1}{2}Eh(D(n|d))$ from $E\text{NS}(n|d)$ gives us $E\text{NS}^*(n|d)$ by definition (see Eq. (4.12)), we therefore obtain $E\text{NS}^*(n|d)$ as an explicit function of n, i.e., the right-hand side of Eq. (4.5) with $h_n(d)$ replaced by $h(d)$.

A similar statement applies to Eqs. (4.7) and (4.8) for $r > 0$, where the term $-\frac{1}{2}h_n(\delta)$ appears implicitly. Here, subtraction of

$$\frac{1}{2}\, Eh(D(n,r|d)) = \frac{1}{2}\, Eh(D(n|\delta))$$

produces explicit functions of n in which $h_n(\delta)$ is replaced by $h(\delta)$.

Under a given policy A, the state $(c,n,r), n,r \geq 1$, is called a *preemption state* if a new job is assigned in that state. The corresponding decision, $f^A(c,n,r)$, is called a preemption. The following lemma proves that NS is optimal for the expected modified flow time, as well as supplying the bound needed for Theorem 4.2.

LEMMA 4.2: *If $p < 1/k$, then for any policy A, we have*

$$E[A^*(\sigma) - \text{NS}^*(\sigma)] \geq \rho^A(\sigma)(1 - pk)\mu, \tag{4.14}$$

where $\rho^A(\sigma)$ is the expected number of preemptions made by A starting in state σ.

PROOF: Based on a given policy A and an initial state $\sigma = (c,n,r)$, we define a sequence of policies $A_1, A_2, \ldots, A_{n+1}$, where $A_1 \equiv A$, $A_{n+1} \equiv \text{NS}$, and A_j, $2 \leq j \leq n$, is identical to A in states at level j or greater, but is identical to NS in states at levels less than j. We write

$$E[A^*(\sigma) - \text{NS}^*(\sigma)] = \sum_{j=1}^{n} E[A_j^*(\sigma) - A_{j+1}^*(\sigma)]. \tag{4.15}$$

To obtain a lower bound on $E[A_j^*(\sigma) - A_{j+1}^*(\sigma)]$, suppose $\sigma_j = (c_j, j, r_j)$ is a preemption state under A, and hence A_j. By definition of A_j and A_{j+1}, policy A_j starts with a preemption in (c_j, j, r_j), but is otherwise non-preemptive, and policy A_{j+1} is simply NS in state σ_j; i.e., $A_{j+1}^*(\sigma_j) = \text{NS}^*(\sigma_j)$.

For the case $1 \le c_j \le k$, and hence $1 \le d = k - c_j + 1 \le k$, we write for A_j

$$EA_j^*(\sigma_j) = EA_j(\sigma_j) - \frac{1}{2}\,Eh(D^{A_j}(\sigma_j)).$$

Since A_j begins with a preemption, we can substitute for $EA_j(\sigma_j)$ as follows:

$$EA_j^*(\sigma_j) = j + r_j + p\mathrm{NS}(j-1, r_j | d-1) + (1-p)E\mathrm{NS}(j-1, r_j+1 | d-1) - \frac{1}{2}\,Eh(D^{A_j}(\sigma_j)). \tag{4.16}$$

Next, by definition of A_j, we can substitute

$$Eh(D^{A_j}(\sigma_j)) = pEh(D(j-1, r_j | d-1)) + (1-p)Eh(D(j-1, r_j+1 | d-1))$$

into Eq. (4.16), and then subtract $E\mathrm{NS}^*(j, r_j | d)$ to obtain

$$\begin{aligned} E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] &= j + r_j + pE\mathrm{NS}^*(j-1, r_j | d-1) \\ &\quad + (1-p)E\mathrm{NS}^*(j-1, r_j+1 | d-1) \\ &\quad - E\mathrm{NS}^*(j, r_j | d). \end{aligned} \tag{4.17}$$

Similarly, for $c_j = \lambda$, we obtain

$$\begin{aligned} E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] &= p[1 + E\mathrm{NS}^*(j-1, r_j | 1)] \\ &\quad + (1-p)[k + 1 + E\mathrm{NS}^*(j-1, r_j | k+1)] \\ &\quad - E\mathrm{NS}^*(j, r_j | 0), \end{aligned} \tag{4.18}$$

and for $c_j = 0$, we have by Eq. (4.1),

$$\begin{aligned} E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] &= p[j + r_j + pE\mathrm{NS}^*(j-1, r_j | 0) \\ &\quad + (1-p)E\mathrm{NS}^*(j-1, r_j+1 | 0)] \\ &\quad + (1-p)[j + r_j + pE\mathrm{NS}^*(j-1, r_j | k) \\ &\quad + (1-p)E\mathrm{NS}^*(j-1, r_j+1 | k)] \\ &\quad - [pE\mathrm{NS}^*(j, r_j | 1) + (1-p)E\mathrm{NS}^*(j, r_j | k+1)]. \end{aligned} \tag{4.19}$$

Using Eqs. (4.5), (4.7), and (4.8) with $h_n(\delta)$ replaced by $h(\delta)$, we can evaluate the right-hand sides of Eqs. (4.17)–(4.19) as explicit functions of the parameters and obtain the uniform bound

$$E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] \ge \mu(1 - pk). \tag{4.20}$$

The calculations, although lengthy, are routine. In the Appendix, we show how the calculations can be done more simply by exploiting the fact that A_j and A_{j+1} differ in only one decision.

Now let $q_j^A(\sigma)$ denote the probability that A, starting in state σ, reaches one of the preemption states (c_j, j, r_j) at level j. Then

$$E[A_j^*(\sigma) - A_{j+1}^*(\sigma)] \geq q_j^A(\sigma)\mu(1 - pk), \qquad 1 \leq j \leq n.$$

and

$$E[A^*(\sigma) - \mathrm{NS}^*(\sigma)] \geq \mu(1 - pk) \sum_{j=1}^{n} q_j^A(\sigma). \tag{4.21}$$

Since $\rho^A(\sigma) = \sum_{j=1}^{n} q_j^A(\sigma)$, Eq. (4.14) follows. ■

It is now easy to prove the main result of this section: Both the expected added cost of NS relative to OPT and the expected number of preemptions made by OPT are bounded by constants independent of the initial state.

THEOREM 4.2: *If $p < 1/k$, then for any initial state σ,*

$$E[\mathrm{NS}(\sigma) - \mathrm{OPT}(\sigma)] \leq (k + 2)/2, \tag{4.22}$$

and

$$\rho^{\mathrm{OPT}}(\sigma) \leq \frac{k + 2}{2\mu(1 - pk)}. \tag{4.23}$$

PROOF: By Eqs. (4.10) and (4.12), we have

$$|E\mathrm{NS}^*(\sigma) - E\mathrm{NS}(\sigma)| \leq (k + 2)/4, \tag{4.24}$$

$$|E\mathrm{OPT}^*(\sigma) - E\mathrm{OPT}(\sigma)| \leq (k + 2)/4. \tag{4.25}$$

From Eq. (4.24) and Lemma 4.2, we obtain

$$\begin{aligned} E\mathrm{NS}(\sigma) &\leq E\mathrm{NS}^*(\sigma) + (k + 2)/4, \\ &\leq E\mathrm{OPT}^*(\sigma) + (k + 2)/4. \end{aligned}$$

Substitution of $E\mathrm{OPT}^*(\sigma) \leq E\mathrm{OPT}(\sigma) + (k + 2)/4$ yields Eq. (4.22).

As above, we can apply Eq. (4.25), the inequality $E\mathrm{OPT}(\sigma) \leq E\mathrm{NS}(\sigma)$, and Eq. (4.24), in that order, to obtain

$$E\mathrm{OPT}^*(\sigma) - E\mathrm{NS}^*(\sigma) \leq (k + 2)/2. \tag{4.26}$$

By Lemma 4.2, we have

$$E\mathrm{OPT}^*(\sigma) - E\mathrm{NS}^*(\sigma) \geq \rho^{\mathrm{OPT}}(\sigma)\mu(1 - pk). \tag{4.27}$$

The bound given by Eq. (4.23) follows directly from Eqs. (4.26) and (4.27). ■

5. CONCLUSIONS

A problem, which is arguably the simplest multimachine flow-time scheduling problem with job running times not having a monotone hazard-rate distribution, has been shown to have a surprisingly rich structure. In particular, it appears that the optimal policy cannot be determined by a "ranking function." Nevertheless, the SERPT rule was proved to be asymptotically optimal in a strong sense: the expected number of OPT decisions that are not SERPT decisions is bounded by a constant independent of the initial state. An even stronger turnpike optimality was proved for the case $p \geq 1/k$. Equally satisfying was the result (see Eqs. (3.3) and (4.22)) that SERPT yields an expected flow time which exceeds the optimal value by a bounded amount, independent of the initial state.

There are several obvious challenges for future research:

1. Prove that property (P2) of Section 1 is not restrictive. This should follow from Bellman equations.
2. Resolve the question of whether a turnpike theorem can be proved for NS as was the case for PS (see Theorem 3.2).
3. Extend the results to 3 or more machines, to the expected makespan objective function, and to two-point distributions, where the larger point is not a multiple of the smaller point.

Acknowledgment

Research of G. Weiss was supported in part by the National Science Foundation Grant ECS 8712798.

References

1. Agrawala, A., Coffman, E.G., Jr., Garey, M.R., & Tripathi, S. (1984). A stochastic optimization algorithm minimizing expected flow times on uniform processors. *IEEE Transactions on Computers* C-33: 351–356.
2. Bruno, J.L., Downey, P.J., & Frederickson, G.N. (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the Association for Computing Machines* 28: 100–113.
3. Chazan, D., Konheim, A.G., & Weiss, B. (1968). A note on time-sharing. *Journal of Combinatorial Theory* 5: 344–369.
4. Coffman, E.G., Jr., Flatto, L., Garey, M.R., & Weber, R.R. (1987). Minimizing expected makespans on uniform processor systems. *Advances in Applied Probability* 19: 177–201.
5. Feller, W. (1971). *An introduction to probability theory and its applications*, Vol. II, 2nd Edition, New York: John Wiley & Sons.
6. Gittins, J.C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B* 14: 148–177.
7. Glazebrook, K.D. (1979). Scheduling tasks with exponential service times on parallel processors. *Journal of Applied Probability* 16: 658–689.

8. Kämpke, T. (1987). On the optimality of static priority policies in stochastic scheduling on parallel machines. *Journal of Applied Probability* 24: 430–448.
9. Meilijson, I. & Weiss, G. (1977). Multiple feedback at a single server station. *Stochastic Processes and Their Applications* 5: 195–205.
10. Papadimitriou, C. (1985). Games against nature. *Journal of Computer and System Sciences* 31: 288–301
11. Ross, S. (1983). *Introduction to stochastic dynamic programming.* New York: Academic Press.
12. Sevcik, K.C. (1974). Scheduling for minimum total loss using service time distributions. *Journal of the Association for Computing Machines* 21: 66–75.
13. Weber, R.R. (1982). Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time. *Journal of Applied Probability* 19: 167–182.
14. Weber, R.R., Varaiya, P., & Walrand, J. (1986). Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flow time. *Journal of Applied Probability* 23: 841–847.
15. Weiss, G. & Pinedo, M. (1980). Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions. *Journal of Applied Probability* 17: 187–202.

APPENDIX

Properties of $\{D_n\}$. Because of our present need for more indices, we will depart slightly from the conventions in Section 4. We use i and j, $0 \le i, j \le k+1$, to denote delay states; and m and n are used as time parameters.

Feller [5] has analyzed the chain $D_{n+1} = |D_n - X_{n+1}|$ when the X_n's are i.i.d. with a strictly positive density on $[0,\infty]$. In our case, X_n is arithmetic, so Feller's formulas do not apply. On the other hand, $\{D_n\}$ is a finite Markov chain, so geometrically fast convergence to stationary distributions is assured. Properties of $\{D_n\}$ can be derived as follows.

The given distribution of a job's running time, X, is $Pr\{X = 1\} = p$ and $Pr\{X = k+1\} = 1 - p$. The chain $\{D_n\}$ has values in $\{0,1,\ldots,k+1\}$, on the assumption that $0 \le D_0 \le k+1$. $\{D_n\}$ is easily seen to be irreducible; for k odd it is aperiodic (and hence ergodic), and for k even it has period 2. The transition matrix $\mathbf{P} = \{p_{ij}\}$ is given by

$$p_{ij} = \begin{cases} p & i = 0, j = 1 \text{ or } i = 1,\ldots,k+1,\ i \ne \dfrac{k}{2} + 1,\ j = i - 1 \\[2ex] 1 - p & i = 0,\ldots,k+1,\ i \ne \dfrac{k}{2} + 1,\ j = k + 1 - i \\[2ex] 1 & k \text{ even},\ i = \dfrac{k}{2} + 1,\ j = \dfrac{k}{2} \\[2ex] 0 & \text{otherwise.} \end{cases} \tag{A1}$$

By inspection, a solution $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{k+1})$ to $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ is given by

$$\pi_i = \begin{cases} \mu/2 & i = 0, \\ (2-p)\mu/2 & i = 1, \\ (1-p)\mu & 2 \le i \le k, \\ (1-p)\mu/2 & i = k+1, \end{cases} \tag{A2}$$

where $1/\mu = EX = k + 1 - kp$ is the mean job-running time. For k odd, the distribution of D_n converges geometrically fast to $\boldsymbol{\pi}$; for k even, the distribution converges geometrically fast to $\boldsymbol{\pi}^{(e)}$, with $\pi_i^{(e)} = 2\pi_i \dfrac{1+\xi_i}{2}$, and to $\boldsymbol{\pi}^{(o)}$, with $\pi_i^{(o)} = 2\pi_i \dfrac{1-\xi_i}{2}$, according as D_n is even or odd, respectively. (Recall that ξ_i is the parity function, $\xi_i = +1$ for i even, and $\xi_i = -1$ for i odd.) By direct calculation, the first moment of $\boldsymbol{\pi}$ is α, as written in Eq. (4.3). The first moments of $\boldsymbol{\pi}^{(e)}$ and $\boldsymbol{\pi}^{(o)}$ work out to be $\alpha - \mu/2$ and $\alpha + \mu/2$, respectively. Hence, for k odd, $ED_n \to \alpha$ as $n \to \infty$, while for k even, $E(D_n | D_0 = d) + \xi_{n+d} \dfrac{\mu}{2} \to \alpha$ as $n \to \infty$; again, these convergences are geometrically fast.

PROOF OF THEOREM 4.1. From the definition of $h_n(d)$ (see Eq. (4.6)), we obtain the recurrence

$$h_n(i) = i - \alpha + \sum_{j=0}^{k+1} p_{ij} h_{n-1}(j). \tag{A3}$$

First, consider k odd. The geometric convergence of ED_n to α as $n \to \infty$ implies a similar convergence of $\sum_{n=0}^{\infty} E(D_n - \alpha | D_0 = d)$ to a value, say $h(d)$. Letting $n \to \infty$ in Eq. (A3), we obtain the equations

$$h(i) = i - \alpha + \sum_{j=0}^{k+1} p_{ij} h(j), \qquad i = 0,1,\ldots,k+1. \tag{A4}$$

To solve Eq. (A4), we define the differences $\Delta(i) = h(i+1) - h(i)$. Exploiting the form of p_{ij} in Eq. (A1), we obtain

$$\Delta(0) = 1 - p\Delta(0) - (1-p)\Delta(k),$$
$$\Delta(i) = 1 + p\Delta(i-1) - (1-p)\Delta(k-i), \qquad 1 \le i \le k,$$

which is solved by

$$\Delta(i) = h(i+1) - h(i) = \frac{\mu}{2}(2i+1), \qquad 0 \le i \le k.$$

It follows that $h(i) - h(0) = \frac{\mu}{2} i^2$, $0 \le i \le k+1$. Also, $\sum_{i=0}^{k+1} \pi_i h_n(i) = 0$ for all n, so that $\sum_{i=0}^{k+1} \pi_i h(i) = 0$. Thus,

$$-h(0) = \sum_{i=0}^{k+1} \pi_i [h(i) - h(0)] = \frac{\mu}{2} \sum_{i=0}^{k+1} \pi_i i^2,$$
$$= \frac{\mu}{12} + \frac{\mu^2}{6} EX^3.$$

Theorem 4.1 follows for k odd.

For k even, $E(D_n | D_0 = d) + \xi_{n+d} \frac{\mu}{2} \to \alpha$ geometrically fast, and therefore

$$h_n(d) + \frac{\mu}{2} \sum_{0 \le m \le n-1} \xi_{m+d}$$

converges geometrically fast, to $g(d)$ say. The sum is 0 for n even, and ξ_d for n odd. Then as $n \to \infty$.

$$h_{2n}(d) \to g(d),$$

$$h_{2n+1}(d) \to g(d) - \frac{\mu}{2} \xi_d.$$

Letting $n \to \infty$ in the recurrence Eq. (A3), and noting that $p_{ij} = 0$ for $\xi_i = \xi_j$, we obtain the equations

$$\left[g(i) - \frac{\mu}{4}\,\xi_i\right] = (i - \alpha) + \sum_{j=0}^{k+1} p_{ij}\left[g(j) - \frac{\mu}{4}\,\xi_j\right], \qquad 0 \le i \le k+1,$$

from which, similar to the analysis for k even, we arrive at

$$g(i) - g(0) = \begin{cases} i^2\,\dfrac{\mu}{2} & i \text{ even}, \\[2ex] (i^2 - 1)\,\dfrac{\mu}{2} & i \text{ odd}. \end{cases}$$

Observing that $\sum_{i=0}^{k+1} \pi_i^{(e)} h_{2n}(i) = 0$ for all n, we have $\sum_{i=0}^{k+1} \pi_i^{(e)} g(i) = 0$, and hence

$$-g(0) = \sum_{i=0}^{k+1} \pi_i^{(e)}[g(i) - g(0)] = \sum_{i=0}^{k+1} \pi_i^{(e)} i^2\,\frac{\mu}{2},$$

$$= \frac{k(k+1)(k+2)}{6}\,\mu^2 = \frac{\mu^2}{6}\,EX^3 - \frac{\mu}{6}.$$

On substitution, we finally obtain $g(i) = h(i) + \xi_i \dfrac{\mu}{4}$ and, as $n \to \infty$,

$$h_{2n}(d) \to h(d) + \xi_d\,\frac{\mu}{4},$$

$$h_{2n+1}(d) \to h(d) - \xi_d\,\frac{\mu}{4}. \qquad \blacksquare$$

Proof of Eq. (4.20) for $1 \le c_j \le k$. We need to prove that

$$\Delta = E[A_j^*(n,r|d) - \mathrm{NS}^*(n,r|d)] \ge \mu(1 - pk), \qquad 1 \le d \le k.$$

The approach below, along with Eq. (4.1), is easily adapted to a proof of Eq. (4.20) for $c_j = \lambda$ and $c_j = 0$; the details are left to the interested reader. Accompanying the results below are figures describing schedules up through the first r tail-job assignments and the first new-job assignment. The dashed tail job is absent with probability p, and present with probability $1 - p$.

Case 1 ($1 \le d < k$): For r even, inspection of Figure A1 shows that

$$\Delta = \frac{r}{2}\,(1 - pk), \; r \text{ even}.$$

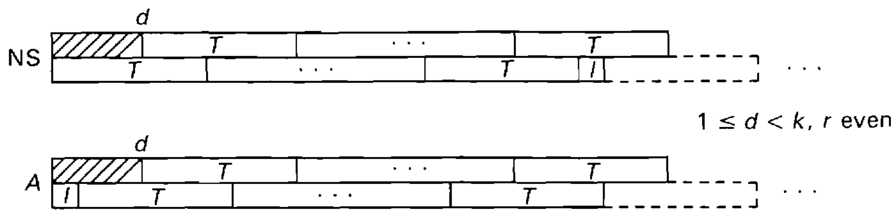


Figure A.1.

For r odd, we obtain with the help of Figure A2,

$$\Delta = \frac{r+1}{2} - \left(\frac{r-1}{2} k + d\right)p - (1-p)$$
$$- p\,\frac{h(k-d+1) - h(k-d-1)}{2} + (1-p)\,\frac{h(d+1) - h(d-1)}{2}.$$

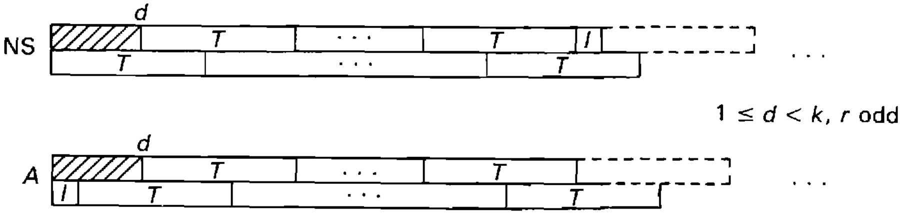


FIGURE A.2.

The last two terms express the difference in the flow time of the last $n-1$ complete jobs; the running of these jobs starts on the average at the same time under both policies. Note that $[h(i+1) - h(i-1)]/2 = \mu i$ and $(1 - \mu k) = \mu(1 - pk)$, so that

$$\Delta = (1-pk)\,\frac{r-1}{2} - pd + p + \mu d - \mu pk,$$
$$= (1-pk)\left[\frac{r-1}{2} + \mu(p + (1-p)d)\right].$$

Case 2 ($d = k$): For r even, we obtain the same result as in Case 1, namely,

$$\Delta = \frac{r}{2}\,(1-pk), \qquad r \text{ even}.$$

For r odd, Figure A3 shows that

$$\Delta = \frac{r+1}{2}\,(1-pk) - (1-p) + (1-p)\,\frac{h(k+1) - h(k-1)}{2},$$
$$= (1-pk)\left[\frac{r+1}{2} - (1-p)\mu\right],$$
$$= (1-pk)\left[\frac{r-1}{2} + \mu(p + (1-p)k)\right], \quad r \text{ odd}.$$

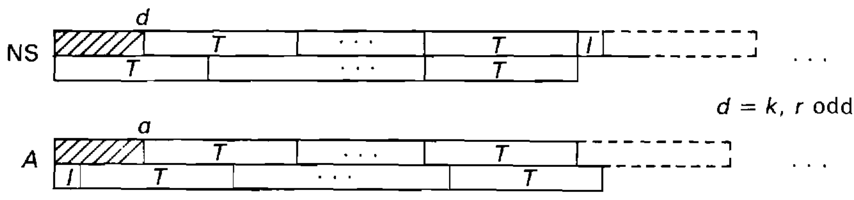


FIGURE A.3.

Probability in the Engineering and Informational Sciences, 2, 1988, 269–278. Printed in the U.S.A.

BRANCHING BANDIT PROCESSES

GIDEON WEISS
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205
and
Department of Statistics
Tel Aviv University

A set of n_i arms of type i, $i = 1, \ldots, L$, is available. A pull of arm of type i occupies a duration V_i at the end of which a reward C_i and $N_{i1}, \ldots, N_{iL}$ new arms are obtained, while all other arms are frozen. A Gittins priority order of types is obtained and shown to yield the maximal discounted reward from this branching process of arms.

1. INTRODUCTION

In the classic multiarmed bandit problem there is a fixed set of N arms, which are in states $X_1(t), \ldots, X_N(t)$. At time t one of the arms, say $i = i(t)$, is chosen and pulled. This yields an immediate reward, $R(t) = R_i(X_i(t))$, and a Markovian transition of arm i to state $X_i(t+1)$, depending on $X_i(t)$ alone. Meanwhile, all other arms are frozen at $X_j(t+1) = X_j(t)$, $j \neq i(t)$. The objective is to choose the arms $i(1), \ldots, i(t), \ldots$ so as to maximize $\left\{\sum_{t=1}^{\infty} \alpha^t R(t)\right\}$. Gittins [2] has shown how to calculate an index $\nu_i = \nu_i(X_i(t))$, which depends on arm i and its state alone, so that the optimal policy is always to choose among $i = 1, \ldots, N$ the arm with the highest index. Many special cases, several extensions, and revisions of the original proof can be found in a series of papers by Gittins and some of his coworkers (Gittins and Glazebrook [4], Gittins and Nash

This research has been partially supported by the National Science Foundation Grant ECS-8712798

[5], Glazebrook [6], and Kelly [11]). Whittle [19] gives a different proof, and extends [20] the results to an "open" multiarmed bandit process, where in any interval $(t, t+1)$ new arms can arrive, in an arrival stream which is independent and identically distributed (i.i.d.), independent of the control $i(s)$ $s \leq t$. Varaiya et al. [18] give very direct proofs to the original theorems of Gittins and to the "open" process. More recently, Chen and Katehakis [1], Kallenberg [9], and Katehakis and Veinott [10] discuss the calculation of the index, whereas Glazebrook [7] presents a sensitivity analysis. In the present paper, we consider a more general "open" process in which the arrivals depend on the arm which is pulled. We follow throughout the approach used in Meilijson and Weiss [14], which contains results on the average cost criterion. The approach is also similar to that of Varaiya et al. [18].

Consider the following model: Arms can be in one of a finite set of possible states, which we call types; let the type of an arm be denoted by I, $I = 1, 2, \ldots, L$. Instead of listing the different arms and their states, the state of the system at any moment is given by $(n_1, \ldots, n_L)$, where n_i is the number of arms of type i present. With an arm of type I, we associate a time V_I, a reward C_I, and arrivals N_{Ij}, $j = 1, \ldots, L$, so that pulling the arm keeps the system busy for a duration V_I, at the end of which a reward C_I is received, and the arm is replaced by an integer number of new arms N_{Ij}, $N_{Ij} \geq 0$, of types j, $j = 1, \ldots, L$. We assume that given I, the durations, the rewards, and the descendants V_I, C_I, N_{Ij}, $j = 1, \ldots, L$, are random variables with arbitrary joint distributions, independent of all other arms, and identically distributed for the same I, $I = 1, \ldots, L$. Starting from some state $(n_1, \ldots, n_L)$, we want to select a type of arm to pull and a policy which will tell us how to continue pulling arms, so as to maximize the sum of discounted expected rewards C_I's obtained over $(0, \infty)$. We shall assume a constant discount rate β, so that a reward C_I received at t is worth $e^{-\beta t} C_I$ now.

In Section 2, we derive some expressions for durations and rewards obtained by following priority rules. In Section 3, we use those to define an index, and prove that the optimal policy is to always pull arms of types with highest index available.

The model of Whittle [20] is a special case of our model in which all the durations V_I are 1, and in which the arrivals N_{Ij} consist of a transition of the arm of type I to an arm of a new type, and of additional arrivals of arms independent of I or of the transition.

2. DISCOUNTED TIMES AND REWARDS OF A PRIORITY POLICY

With an arm of type $i (i = 1, \ldots, L)$ is associated a reward C_i, a nonnegative duration V_i, and a vector of integers N_{ij} of the number of descendants of the arm, of types $j = 1, \ldots, L$. The joint distribution of V_i, N_{ij}, $j = 1, \ldots, L$, is characterized by

$$g_i(s, Z_1, \ldots, Z_L) = E(e^{-sV_i} Z_1^{N_{i1}} \cdots Z_L^{N_{iL}}), \qquad i = 1, \ldots, L. \tag{1}$$

Throughout this section, we shall assume that $1, \ldots, L$ is a priority order, so that if the state at a decision moment is $(n_1, \ldots, n_L)$, the arm chosen is of type

$$i = \min\{j \mid n_j > 0\}.$$

A typical sequence of decisions will start by pulling an arm of type i; for the sake of definiteness, we will assume arms are ordered within types and choose the latest arm of type i. The next decision will occur a time V_i later, from a new state $(N_{i1}, \ldots, N_{ii-1}, n_i - 1 + N_{ii}, n_{i+1} + N_{ii+1}, \ldots, n_L + N_{iL})$. We assume the $N_{i1}, \ldots, N_{iL}$ new arms are again ordered and are all later than the original $n_1, \ldots, n_L$ arms, and when choosing an arm of any type, the latest is chosen (a LIFO policy, within types). If any arms of type 1 were created, the latest is pulled. Arms of type 1 (if there are any) will then be pulled in succession, choosing always the latest one, and creating new arms, for as long as any arms of type 1 are present, or indefinitely. When no more arms of type 1 are present, the latest arm of type 2 is pulled (if there is any), followed again by pulls of arms of type 1; the entire birth process of each arm of type 1 is cleared, moving on to earlier arms until again no more arms of type 1 remain, when the next arm of type 2, the latest created, is pulled. Following the pull of the latest of n_k arms of type k, arms of lower types are pulled until the whole birth process of arms of types $1, \ldots, k-1$ is cleared; then a pull of the latest arm of type k and all its descendants of types $1, \ldots, k-1$ follows, until there are $n_k - 1$ arms of type k left. Only when all arms of type k, and descendants of types $1, \ldots, k$ are cleared, can we move to the latest (if any) arm of type $k+1$, etc.

Consider an ordered pair of types (i,j), where at time $t = 0$ the state consists of a single arm of type i, and consider the sequence of arm pulls starting with the arm of type i, followed by arm pulls of types $1, \ldots, j$ until all arms of types $1, \ldots, j$ are exhausted for the first time, or indefinitely. Call this an (i,j) period. Let T_{ij} be the duration (possibly infinite) of an (i,j) period, and let $M_{ijk}, k = 1, \ldots, L$, be the number of type k arms present at the end of the (i,j) period, where of course $M_{ijk} = 0$, $k \le j$, and where all these arms were created during the (i,j) period. The joint distribution of T_{ij}, M_{ijk}, $k = 1, \ldots, L$, is characterized by

$$G_{ij}(s, Z_1, \ldots, Z_L) = E(e^{-sT_{ij}} Z_1^{M_{ij1}} \cdots Z_L^{M_{ijL}}). \tag{2}$$

Define also

$$G_{io}(s, Z_1, \ldots, Z_L) = g_i(s, Z_1, \ldots, Z_L). \tag{3}$$

Proposition 1: The generating functions G_{ij} satisfy the recursive relation

$$G_{ij}(s, Z_1, \ldots, Z_L) = G_{ij-1}(s, Z_1, \ldots, Z_{j-1}, G_{jj}(s, Z_1, \ldots, Z_L), Z_{j+1}, \ldots, Z_L).$$

(4)

PROOF: The (i,j) period of duration T_{ij} will consist of an $(i,j-1)$ period of duraiton T_{ij-1}, at the end of which there will be M_{ij-1j} type j arms; conditional on $M_{ij-1j} = m$, these give rise to m i.i.d. (j,j) periods. Writing this out we get

$$G_{ij}(s,Z_1,\ldots,Z_L) = E\left[e^{-sT_{ij-1}} \prod_{k>j} Z_k^{M_{ij-1k}} \left\{E\left(e^{-sT_{jj}} \prod_{l>j} Z_l^{M_{jjl}}\right)\right\}^{M_{ij-1j}}\right]$$

$$= G_{ij-1}(s,1,1,\ldots,1,G_{jj}(s,1,\ldots,1,Z_{j+1},\ldots,Z_L), Z_{j+1},\ldots,Z_L)$$

which is the required result. ■

The relations (3) and (4) enable us in principle to obtain the generating functions G_{ij} recursively: given G_{ij-1} for all i, we need to solve

$$G_{jj}(s,Z_1,\ldots,Z_L) = G_{jj-1}(s,Z_1,\ldots,Z_{j-1},G_{jj}(s,Z_1,\ldots,Z_L),Z_{j+1},\ldots,Z_L), \tag{5}$$

and then obtain G_{ij} for all i by substituting in Eq. (4).

The random variables T_{ij} and M_{ijk}, $k = 1,\ldots,L$, give the duration of the (i,j) period and also the state at the end of the (i,j) period. Let β be the discount rate, let $C_1,\ldots,C_L$ be the rewards of arms of types $1,\ldots,$L, and let W_{ij} be the total discounted reward accumulated in the (i,j) period. Define also

$$\gamma_{ij} = E(e^{-\beta T_{ij}}) = G_{ij}(\beta,1,\ldots,1), \qquad \gamma_{io} = E(e^{-\beta V_i}), \tag{6}$$

$$d_{ij} = E(W_{ij}), \qquad d_{io} = E(W_{io}) = E(C_i e^{-\beta V_i}). \tag{7}$$

Proposition 2: The expected discounted reward for period (i,j) satisfies the relations

$$d_{ij} = d_{ij-1} + \frac{d_{jj}}{1-\gamma_{jj}}(\gamma_{ij-1} - \gamma_{ij}), \tag{8}$$

$$d_{jj} = d_{jj-1}\frac{1-\gamma_{jj}}{1-\gamma_{jj-1}}. \tag{9}$$

PROOF: As in the proof of Proposition 1, decompose the (i,j) period into an $(i,j-1)$ period followed by M_{ij-1j} conditionally independent (j,j) periods, which we shall denote by superscripts $1,\ldots,M_{ij-1j}$. Let W_{ij-1} and $W_{jj}^{(1)},\ldots,W_{jj}^{(M_{ij-1j})}$ be the discounted rewards of these periods, discounted to the beginning of these periods. We can write

$$W_{ij} = W_{ij-1} + e^{-\beta T_{ij-1}}\{W_{jj}^{(1)} + e^{-\beta T_{jj}^{(1)}} W_{jj}^{(2)} + \cdots + e^{-\beta(T_{jj}^{(1)} + \cdots + T_{jj}^{(M_{ij-1j}-1)})} W_{jj}^{(M_{ij-1j})}\},$$

We note that, conditional on the value of M_{ij-1j}, the pairs $T_{jj}^{(k)}, W_{jj}^{(k)}$ are independent of T_{ij-1}, W_{ij-1}, and are independent of each other for $k = 1, \ldots, M_{ij-1j}$, and are identically distributed with the same joint distribution as the generic pair T_{jj}, W_{jj}. Therefore,

$$\begin{aligned}
d_{ij} &= d_{ij-1} + E[e^{-\beta T_{ij-1}} E\{W_{jj}^{(1)} + e^{-\beta T_{jj}^{(1)}} W_{jj}^{(2)} \\
&\quad + e^{-\beta(T_{jj}^{(1)} + \cdots + T_{jj}^{(M_{ij-1j}-1)})} W_{jj}^{(M_{ij-1j})} | M_{ij-1j}\}], \\
&= d_{ij-1} + E[e^{-\beta T_{ij-1}} d_{jj}(1 + \gamma_{jj} + \cdots + \gamma_{jj}^{M_{ij-1j}-1})], \\
&= d_{ij-1} + \frac{d_{jj}}{1-\gamma_{jj}} E\{e^{-\beta T_{ij-1}}(1 - \gamma_{jj}^{M_{ij-1j}})\}, \\
&= d_{ij-1} + \frac{d_{jj}}{1-\gamma_{jj}} \{\gamma_{ij-1} + G_{ij-1}(\beta,1,\ldots,G_{jj}(\beta,1,\ldots,1),1,\ldots,1)\}, \\
&= d_{ij-1} + \frac{d_{jj}}{1-\gamma_{ij}} (\gamma_{ij-1} - \gamma_{ij}).
\end{aligned}$$

Substituting j for i in Eq. (8) and solving, we obtain Eq. (9). ■

We now define the (i,j) priority index I_{ij} as

$$I_{ij} = \beta \frac{d_{ij}}{1-\gamma_{ij}}. \tag{10}$$

Note that

$$(1-\gamma_{ij})/\beta = \{1 - E(e^{-\beta T_{ij}})\}/\beta = E\left(\int_0^{T_{ij}} e^{-\beta t} dt\right).$$

Thus, the index I_{ij} has the typical structure of a Gittins index, in that it is a ratio of expected discounted reward over expected discounted time, calculated for the duration T_{ij}, and relative to the priority order $1, \ldots, L$.

Using Eqs. (8) and (9), we get

$$I_{jj} = I_{jj-1}, \tag{11}$$

$$I_{ij} = \frac{1-\gamma_{ij-1}}{1-\gamma_{ij}} I_{ij-1} + \left(1 - \frac{1-\gamma_{ij-1}}{1-\gamma_{ij}}\right) I_{jj-1}. \tag{12}$$

In Eqs. (11) and (12), we see that I_{ij} is a convex combination of I_{ij-1} and I_{jj} (or I_{jj-1}). In other words, the reward per unit time in period (i,j) is a weighted average of the reward per unit time in the initial $(i,j-1)$ period and the rewards of the (j,j) periods which follow it. We can reiterate Eq. (12) to get

$$\begin{aligned}
I_{ij} &= \alpha_1 I_{10} + \alpha_2 I_{20} + \cdots + \alpha_j I_{j0} + (1 - \alpha_1 \cdots - \alpha_j) I_{i0}, \\
&= \sum_{k=1}^{j} \alpha_k \frac{\beta E(C_k e^{-\beta V_k})}{1 - E(e^{-\beta V_k})} + (1 - \alpha_1 \cdots - \alpha_j) \frac{\beta E(C_i e^{-\beta V_i})}{1 - E(e^{-\beta V_i})}.
\end{aligned} \tag{13}$$

3. THE GITTINS PRIORITY ORDER AND THE PROOF OF OPTIMALITY

In Section 2, we have taken an arbitrary priority order and defined an index I_{ij} for each pair of types. We now define the optimal priority order, which we call the Gittins order.

DEFINITION: *The Gittins priority order of types is defined by*

$$I_{11} = I_{10} = \max_{1 \le i \le L} I_{i0},$$

$$I_{jj} = I_{jj-1} = \max_{j \le i \le L} I_{ij-1}. \tag{14}$$

Although the definition given by Eq. (14) appears to be implicit, it is obvious that the order can be calculated recursively, where type 1 is obtained by calculating all the I_{i0} and choosing the maximal, and, having defined types $1, \ldots, j-1$, type j is obtained by calculating the values of I_{ij-1} for all i different from $1, \ldots, j-1$, and choosing the maximal; a tie-breaking mechanism for several maxima can be chosen arbitrarily.

We also define the Gittins index of type j as

$$I_j^* = I_{jj}. \tag{15}$$

THEOREM 1: *The priority policy which uses the Gittins priority ordering is optimal among all policies.*

PROOF: Our branching bandit process problem is a semi-Markov decision problem, formulated as follows. The state is the vector of numbers of arms of various types:

$$s = (n_1, \ldots, n_L).$$

The actions available at state s are

$$J(s) = \{i \mid n_i > 0\}.$$

The transition from state s under action i will involve a duration

$$T(s,i) = V_i$$

and a new state

$$s' = (n_1 + N_{i1}, \ldots, n_i - 1 + N_{ii}, \ldots, n_L + N_{iL}).$$

A reward C_i will be obtained at the end of $T(s,i)$. All rewards are discounted with continuous discount rate $\beta > 0$.

Consider a general strategy π; let $\pi(s)$ be the process of states and decisions induced by π, for an initial state s, and let $V(\pi)(s)$ be the total expected discounted reward under π, for initial stage s. A nonrandomizing decision function f is a function from the states into the available actions, $f(s) \in J(s)$. A stationary policy is defined by a decision function f, if at all decision moments it chooses $f(s)$; we note that policy by f^∞, and the process it generates for initial state s by $f^{\infty}(s)$. T is a stopping time for the stationary policy f^{∞}, if for every initial state s, T is a stopping time of $f^{\infty}(s)$. We denote by $(f^{(T)}, \pi)$ the policy which does f up to T, and continues with π.

We wish to find an optimal policy π^* such that $V(\pi^*)(s) = \sup_{\pi} V(\pi)(s)$.

From the theory of Markov decision processes (Ross [16]), because the action space is finite, we know that there exists a stationary optimal policy. Furthermore, f^{∞} is optimal if it is excessive, that is, for every decision function g and initial state s,

$$V(g,f^{\infty})(s) \leq V(f^{\infty})(s). \tag{16}$$

To prove excessivity it is enough to proceed as follows: for any g and s_o, if $g(s_o) = f(s_o)$ there is nothing to prove. If $g(s_o) = i \neq f(s_o)$, look at

$$h(s) = \begin{cases} i & i \in J(s) \\ f(s) & \text{otherwise} \end{cases}$$

so that $h(s_o) = g(s_o) = i$. Let T be some (possibly infinite) stopping time, $E(T) > 0$, defined on f^{∞}. To show Eq. (16) for s_o, it is enough to show that for all s

$$V(h,f^{\infty})(s) \leq V(f^{(T)},h,f^{\infty})(s), \tag{17}$$

since iterating this m times gives

$$V(h,f^{\infty})(s) \leq V(f^{(T)m},m,f^{\infty})(s) \underset{n\to\infty}{\to} V(f^{\infty})(s). \tag{18}$$

We now let $1,\ldots,L$ be the Gittins priority order, and let f be the priority decision function

$$f(s) = \min(k \mid n_k > 0). \tag{19}$$

We choose as T the stopping time $T_{f(s)f(s)}$ on $f^{\infty}(s)$; clearly $E(T) \geq \min E(V_k) > 0$.

We now fix i, and show that Eq. (17) holds. If $h(s) = f(s)$ there is nothing to show; otherwise, the state s is

$$\begin{gathered} s = (0,\ldots,0,n_j,\ldots,n_i,\ldots), \qquad n_i n_j > 0, \\ f(s) = j < i = h(s). \end{gathered} \tag{20}$$

We now compare $V(h,f^{\infty})(s)$ to $V(f^{(T)},h,f^{\infty})(s)$. The two processes generated by the policies are described first; see Figure 1.

The policy $(h,f^{\infty})(s)$ will act as follows: it will choose the latest arm of type i, then all its descendants of types $\leq j$; this will take time T_{ij} with resulting state

$$s^h = (0,\ldots,0,n_j,n_{j+1} + M_{ijj+1},\ldots,n_i - 1 + M_{iji},\ldots).$$

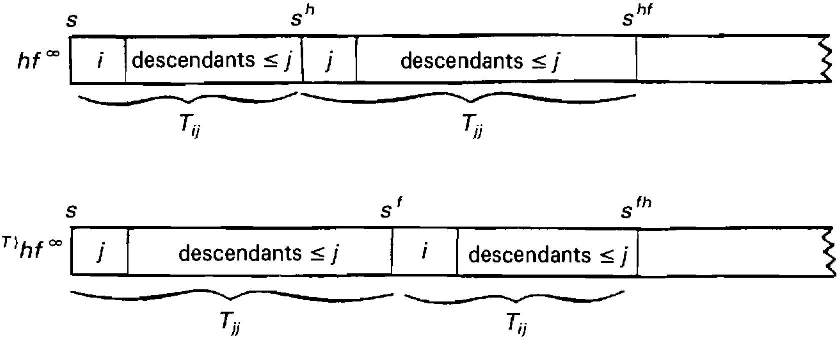


FIGURE 1. Comparison of two policies.

It will then choose the latest original type j arm, followed by its descendants of types $\le j$; this will take an additional time T_{jj} with resulting state

$$s^{hf} = (0,\ldots,0,n_j - 1, n_{j+1} + M_{ijj+1} + M_{jjj+1},\ldots,n_i - 1 + M_{iji} + M_{jji},\ldots).$$

From time $T_{ij} + T_{jj}$ and state s^{hf} onwards, the policy will act as f^∞.

The policy $(f^{(T)},h,f)(s)$ will act as follows: it will choose the latest type j arm, and continue f for time T; that is, it will choose all descendants of the j arm of types $\le j$, with a duration T_{jj} and resulting state

$$s^f = (0,\ldots,0,n_j - 1, n_{j+1} + M_{jjj+1},\ldots,n_i + M_{jji},\ldots).$$

It will then do h, that is, choose a type i arm, and proceed with f^∞; that is, choose all descendants of i of types $\le j$, with duration T_{ij} and resulting state

$$s^{fh} = (0,\ldots,0,n_j - 1, n_{j+1} + M_{jjj+1} + M_{ijj+1},\ldots,n_i - 1 + M_{jji} - M_{iji},\ldots).$$

From time $T_{jj} + T_{ij}$ and state s^{fh} onwards, it will act as f^∞. Of course, the durations $T_{ij} + T_{jj}$, $T_{jj} + T_{ij}$ and the resulting states s^{hf}, s^{fh} under the two policies have the same distribution. The total discounted rewards of these policies (where we let W denote the random reward of a policy) are:

$$\begin{aligned} W(h,f^\infty)(s) &= W_{ij} + e^{-\beta T_{ij}} W_{jj} + e^{-\beta(T_{ij}+T_{jj})} W(f^\infty)(s^{hf}), \\ W(f^{(T)},h,f^\infty)(s) &= W_{jj} + e^{-\beta T_{jj}} W_{ij} + e^{-\beta(T_{jj}+T_{ij})} W(f^\infty)(s^{fh}). \end{aligned} \tag{21}$$

Taking expectations (with obvious use of independence),

$$\begin{aligned} V(h,f^\infty)(s) &= d_{ij} + \gamma_{ij} d_{jj} + E, \\ V(f^{(T)},h,f^\infty)(s) &= d_{jj} + \gamma_{jj} d_{ij} + E, \end{aligned} \tag{22}$$

where E is the expected value of the last term in Eq. (21), which is the same for both policies. We get (using Eqs. (9), (11), (12), and (14)

$$
\begin{aligned}
V(f^{(T)},h,f^{\infty})(s) - V(h,f^{\infty})(s) &= d_{jj} + \gamma_{jj} d_{ij}, \\
-d_{ij} - \gamma_{ij} d_{jj} &= (1 - \gamma_{ij}) d_{jj} - (1 - \gamma_{jj}) d_{ij}, \\
&= (1/\beta)(1 - \gamma_{ij})(1 - \gamma_{jj})(I_{jj} - I_{ij}), \\
&= (1/\beta)(1 - \gamma_{jj})(1 - \gamma_{ij-1}) \\
&\quad \times (I_{jj-1} - I_{ij-1}) > 0.
\end{aligned}
$$

Note that in the above proof in Eq. (21), we allow the possibility of $E(T_{jj}) = \infty$, or $P(T_{jj} < \infty) < 1$, or even $P(T_{jj} < \infty) = 0$. This does not affect the proof at all, since $\gamma_{jj} = E(e^{-\beta T_{jj}})$ is still well-defined, with the possibility of $\gamma_{jj} = 0$; as long as $E(T_{jj}) > 0$, $1 - \gamma_{jj} > 0$ so I_{jj} is well-defined; of course, if $\gamma_{jj} = 0$, $I_{jj} = \beta d_{jj}$.

References

1. Chen, Y.R. & Katehakis, M.N. (1986). Linear programming for finite state multiarmed bandit problems. *Mathematics of Operations Research* 11: 180–183.
2. Gittins, J.C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B* 14: 148–177.
3. Gittins, J.C. & Jones, D.M. (1974). A dynamic allocation index for the sequential design of experiments. In J. Gani, K. Sarkadi, & I. Vince (eds.), *Progress in Statistics European Meeting of Statisticians 1972, Vol. 1.* Amsterdam: North-Holland, pp. 241–266.
4. Gittins, J.C. & Glazebrook, K.D. (1977). On Bayesian models in stochastic scheduling. *Journal of Applied Probability* 14: 556–565.
5. Gittins, J.C. & Nash, P. (1977). Scheduling, queues, and dynamic allocation indices. *Proceedings EMS, Prague*, 1974, Czechoslovak Academy of Sciences, Prague, 191–202.
6. Glazebrook, K.D. (1976). Stochastic scheduling with order constraints. *International Journal of Systems Science* 7: 657–666.
7. Glazebrook, K.D. (1987). Sensitivity analysis for stochastic scheduling problems. *Mathematics of Operations Research* 12: 205–223.
8. Harrison, J.M. (1975). Dynamic scheduling of a multiclass queue: discount optimality. *Operations Research* 23: 270–282.
9. Kallenberg, L.C.M. (1986). A note on M.N. Katehakis and Y.R. Chen's computation of the Gittins Index. *Mathematics of Operations Research* 11: 184–186.
10. Katehakis, M.N. & Veinott, A.F. (1987). The multiarmed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12: 262–268.
11. Kelly, F.P. (1981). Multiarmed bandits with discount factor near one: the Bernoulli case. *The Annals of Statistics* 9: 987–1001.
12. Klimov, G.P. (1974). Time sharing service systems I. *Theory of Probability and Applications* 19: 532–551.
13. Klimov, G.P. (1978). Time sharing service systems II. *Theory of Probability and Applications* 23: 314–321.
14. Meilijson, I. & Weiss, G. (1977). Multiple feedback at a single server station. *Stochastic Processes and their Applications* 5: 195–205.
15. Nash, P. & Gittins, J.C. (1977). A Hamiltonian approach to optimal stochastic resource allocation. *Advances in Applied Probability* 9: 55–68.

16. Ross, S.M. (1970). *Applied probability models with optimization applications.* San Francisco, CA: Holden Day.
17. Sevcik, K.C. (1974). Scheduling for minimum total loss using service time distributions. *Journal of the Association for Computing Machine* 21: 66–75.
18. Varaiya, P., Walrand, J., & Buyukkoc, C. (1985). Extensions of the multiarmed bandit problem: the discounted case. *IEEE Transactions on Automatic Control* AC-30: 426–436.
19. Whittle, P. (1980). Multiarmed bandits and the Gittins index. *Journal of the Royal Statistical Society Series B* 42: 143–149.
20. Whittle, P. (1981). Arm acquiring bandits. *Annals of Probability* 9: 284–292.

THE "LARGEST VARIANCE FIRST" POLICY IN SOME STOCHASTIC SCHEDULING PROBLEMS

MICHAEL PINEDO

Columbia University, New York, New York

GIDEON WEISS

Georgia Institute of Technology, Atlanta, Georgia

(Received June 1985; revision received August 1986; accepted February 1987)

We consider a situation in which n jobs, requiring random amounts of processing, all with the same mean, are to be scheduled on m parallel machines with respect to one of two objectives: expected flowtime and expected makespan. We discuss optimality of the rule that says to schedule the jobs with the largest variance first (LVF). We show that for some very simple job length distributions, LVF minimizes both the expected flowtime and the expected makespan.

[n trying to solve scheduling problems with jobs whose processing times are not known precisely ı advance but are drawn from given probability istributions, researchers have concentrated on rules ısed on the jobs' mean processing times. One natural ɔproach is to use the expected values as stochastic ırrogates for the exactly known processing times of :terministic jobs. Yet an essential role in stochastic heduling is played by the variability that such jobs troduce into the schedule: this problem feature is :tter described by the variances than by the means ˋ the jobs. In this paper, we investigate scheduling les based on the variance of the distributions from hich the processing times of the jobs are drawn.

Jobs 1, ..., n are to be processed without pre-ɑptions on m parallel identical machines; job j re-ıires processing for a duration X_j on one machine; , ..., X_n are independent random variables. Jobs ll be completed at time $C_1, \ldots, C_n$, which depend ı the X_j's as well as on the scheduling rule. Two mmonly used objective functions for this model are : flowtime $\sum C_j$ and the makespan $C_{\max} = \max C_j$, d the resulting problem is to find schedules that .nimize the expected value of either function. As is own, the longest-expected-processing-time-first ap-ɔach (LEPT) minimizes expected $C_{\max}$ for exponen-l jobs or for remainders of independent, identically :tributed (i.i.d.) decreasing hazard rate jobs (Bruno, ›wney and Frederickson 1981, Pinedo and Weiss 80, and Weber 1982. In general, LEPT is a good t not optimal heuristic. Recently, in a remarkable ɔer, Weber, Varaiya and Walrand (1986) showed that the shortest-expected-processing-time-first approach (SEPT) minimizes expected flowtime, $\sum C_j$, under the very general condition that the job processing times are stochastically comparable (stochastic comparison is defined by $X \leqslant_{\mathrm{ST}} Y$ if $P(X > x) \leqslant P(Y > x)$ for all x); this result generalizes previous work on the problem by Glazebrook (1979), Bruno, Downey and Frederickson, Weiss and Pinedo (1980) and Weber.

In this paper we will look at the case in which all jobs have equal mean processing times, $\mathrm{E}(X_1) = \ldots = \mathrm{E}(X_n) = 1$. In this case SEPT and LEPT are of course meaningless, and we need to look for other features to compare jobs and to improve our intuitive understanding of the problem.

For $\mathrm{E}(X_j) = 1$ for $j = 1, \ldots, n$ we note, first, that, if we have a single machine, then $\mathrm{E}(\sum C_j) = n(n+1)/2$ and $\mathrm{E}(C_{\max}) = n$, independent of the schedule. Second, processing times are not stochastically comparable unless they are identically distributed. With equal means we naturally turn to the variances, and we find heuristic indications that the rule of LVF, largest variance first, provides good schedules for both objectives—expected flowtime and expected makespan.

The reasoning is as follows: In minimizing makespan, we are concerned about the possibility of being stuck with some long jobs at the end of the process, when some machines are idle and our total processing rate is lower than m; this situation seems to be more likely when the more variable jobs are left to be done last, so LVF is attractive. To minimize flowtime, our

ɔject classification: 592 priority scheduling by variance of stochastic processing time.

0030-364X/87/3506-0884 $01.25

main concern in scheduling earlier jobs is to prevent delays to later jobs, of which there are a large number at the start of the schedule. If we schedule the m jobs with the largest variances first, we are likely to have the first job completion as early as possible, so LVF is attractive.

We do not know how widely applicable this heuristic is and in what situations it might supply the optimal policy. In this paper we examine some particularly simple special cases for which LVF is indeed optimal for makespan as well as flowtime. We chose these special cases because they lend themselves to easy analysis, rather than because of their practical importance. Nevertheless, we feel that they provide a strong case for the use of a LVF heuristic when the mean processing times are equal or nearly equal. (Pinedo and Wie 1984 discuss the performance of LVF for machines in series.)

We consider three families of distributions.

Class I

$$F_j(x) = 1 - (1 - 2p_j)e^{-x} - p_j e^{-x/2} \qquad j = 1, \ldots, n. \quad (1)$$

Class II

$$F_j(x) = 1 - (1 - p_j)e^{-x} - p_j x e^{-x} \qquad j = 1, \ldots, n. \quad (2)$$

Class III

$$\begin{aligned} P(X_j = 0) &= p_j \\ P(X_j = 1) &= 1 - 2p_j \qquad j = 1, \ldots, n. \quad (3) \\ P(X_j = 2) &= p_j \end{aligned}$$

In Sections 2 and 3, we will prove the following theorems.

Theorem 1. *Assume $X_1, \ldots, X_n$ belong to one of the classes, I, II, or III. For $m = 2$ machines, LVF minimizes expected flowtime.*

Theorem 2. *Assume $X_1, \ldots, X_n$ belong to one of the classes I or II, with any number of machines m, or to class III with $m = 2$ machines. Then LVF minimizes expected makespan.*

Each distribution in class I, II, and III is a mixture of 0 and the random variables Y and Z, in the proportions p_j, $1 - 2p_j$ and p_j, if we assume that $E(Y) = 1$, $E(Z) = 2$ and for the three classes we have the following situations.

Class I: $Y \sim \exp(1)$, $Z \sim \exp(1/2)$.

Class II: $Y \sim \exp(1)$, $Z \sim \exp(1) * \exp(1)$(i.e., convolution of two exponentials with mean 1).

Class III: $Y = 1$, $Z = 2$.

Thus $E(X_j) = 1$ for all j, and the variance $V(X_j)$ is increasing in p_j, specifically

$$V(X_j) = \begin{cases} 1 + 4p_j & \text{class I} \\ 1 + 2p_j & \text{class II} \\ 2p_j & \text{class III.} \end{cases} \quad (4)$$

We shall refer to mixtures of 0, Y, Z in the general proportions $1 - \alpha - \beta \geq 0$, $\alpha \geq 0$, $\beta \geq 0$ as class I′, II′, III′.

In Section 4 we discuss three topics: preemptive scheduling of jobs of classes I–III, monotonicity of $E(\sum C_j)$ and $E(C_{\max})$ with the parameters p_j, and robustness of the results under wider classes of distributions. We indicate how the optimality of LVF extends to generalizations of both class I and II distributions. On the other hand, LVF is not always optimal: in Section 5 we consider a generalization of class III for which it is sometimes optimal to alternate jobs with large and with small variances in the schedule.

2. Minimization of Expected Flowtime

In this section we prove Theorem 1. We also obtain as Corollary 1, Theorem 2 for the case $m = 2$. There are $m = 2$ machines that become available at times $-D_0/2$, $D_0/2$. The jobs $1, \ldots, n$ have processing times $X_1, \ldots, X_n$, all of which belong to one of the classes I, II, III, with mixing parameters $p_1, \ldots, p_n$. We assume that D_0 is a random variable belonging to the corresponding class I′, II′, III′. The jobs are started in the order $1, \ldots, n$, with completion times $C_1, \ldots C_n$. We shall see how the expected flowtime $E(\sum C_k)$ depends on $p_1, \ldots, p_n$ and show that of all possible permutations of the values of $p_1, \ldots, p_n$ the one with $p_1 \geq p_2 \ldots \geq p_n$ minimizes the expected flowtime. This result will prove the optimality of LVF among all permutation rules, and by using induction and the initial value, D_0, will prove the optimality of LVF among all the nonpreemptive scheduling policies.

Consider the schedule of jobs $1, \ldots, k$ and let $U_k \leq V_k$ be the times at which the two machines finish to process all these jobs. Also, let $D_k = V_k - U_k$; then $U_k + V_k = X_1 + \ldots + X_k$, job $k + 1$ starts at U_k, and we have:

$$C_{k+1} = \frac{X_1 + \ldots + X_k - D_k}{2} + X_{k+1} \qquad k = 0, \ldots, n - 1, \quad (5)$$

$$D_{k+1} = | X_{k+1} - D_k | \qquad k = 0, \ldots, n - 1. \quad (6)$$

'he flowtime is

$$\sum_{=1}^{n} C_k = \frac{n+1}{2} X_1 + \frac{n}{2} X_2$$

$$+ \ldots + \frac{2}{2} X_n - (D_0 + \ldots + D_{n-1})/2 \qquad (7)$$

nd, since $E(X_j) = 1$ for $j = 1, \ldots, n$,

$$\left(\sum_{k=1}^{n} C_k\right) = \frac{n(n+3)}{4} - \frac{1}{2} E\left(\sum_{k=0}^{n-1} D_k\right), \qquad (8)$$

› to minimize expected flowtime we want to maxi- ıize the expectation of $\sum_{k=0}^{n-1} D_k$.

It is easy to verify from (6) that if D_0 belongs to ass I′, II′ or III′ and X_j for $j = 1, \ldots, n$ belong to .e corresponding class I, II or III, then D_k for $k = 1$, ., n belong to the same class I′, II′ or III′ with ixing probabilities $1 - \alpha_k - \beta_k$, α_k, β_k. For classes , II′, or III′, define χ_k and δ_k as 0, 1, 2 according to nether X_k and D_k are distributed as 0 or Y or Z, spectively, Note that

$$D_k) = E(E(D_k \mid \delta_k)) = E(\delta_k) = \alpha_k + 2\beta_k. \qquad (9)$$

'e will now treat each class separately.

ass I. Here D_k and X_{k+1} are each a mixture of ponential random variables with means 0, 1, 2. ;phrase (6) as

$$_{k+1} = \max(D_k, X_{k+1}) - \min(D_k, X_{k+1}). \qquad (10)$$

;call that if Y, Z are exponential with rates λ_1, λ_2 ıd means $1/\lambda_1$, $1/\lambda_2$) then $\min(Y, Z)$ is exponential th rate $\lambda_1 + \lambda_2$, and $\max(Y, Z) - \min(Y, Z)$ independent of $\min(Y, Z)$ and is distributed as $p(\lambda_1)$ with probability $\lambda_2/(\lambda_1 + \lambda_2)$ and as $\exp(\lambda_2)$ th probability $\lambda_1/(\lambda_1 + \lambda_2)$. Conditional on the lue of χ_{k+1} and δ_k we find that β_{k+1}, the value of $\delta_{k+1} = 2$), is

$$_{+1} = \begin{array}{c|ccc} & & \chi_{k+1} & \\ \delta_k & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2/3 \\ 2 & 1 & 2/3 & 1 \end{array} . \qquad (11)$$

;o, for $\delta_k = 0$ to hold it is necessary that $\delta_0 = \chi_1 =$ $= \chi_k = 0$, so, letting $p_0 = 1 - \alpha_0 - \beta_0$,

$$- \alpha_k - \beta_k = P(\delta_k = 0) = p_0 p_1 \ldots p_k. \qquad (12)$$

Combining (11) and (12), we obtain a difference equation

$$\beta_{k+1} = 2/3\, \beta_k + 2/3\, p_{k+1} + 1/3\, p_0 \ldots p_{k+1}, \qquad (13)$$

which is solved by

$$\beta_k = \left(\frac{2}{3}\right)^k \beta_0 + \frac{1}{3} \sum_{j=1}^{k} (2p_j + p_0 \ldots p_j)\left(\frac{2}{3}\right)^{k-j}$$

$$k = 1, \ldots, n. \qquad (14)$$

Substituting (14) in

$$E(D_k) = 1 - p_0 \ldots p_k + \beta_k \qquad (15)$$

and adding, we obtain

$$E\left(\sum_{k=0}^{n-1} D_k\right) = n + 3\left(1 - \left(\frac{2}{3}\right)^n\right)\beta_0 - p_0 + 2 \sum_{k=1}^{n-1} p_k$$

$$- \sum_{k=1}^{n-1} (2p_k + p_0 \ldots p_k)\left(\frac{2}{3}\right)^{n-k}. \qquad (16)$$

Consider now a pairwise exchange between p_k and p_{k+1}, $1 \leqslant k \leqslant n-1$. Let Δ denote the value of (16) minus the expected value when p_k and p_{k+1} are exchanged. Then

$$\Delta = (2/3)^{n-k}(1 - p_0 \ldots p_{k-1})(p_k - p_{k+1})$$

$$k = 1, \ldots, n-1. \qquad (17)$$

We see that $\Delta < 0$ if $p_{k+1} > p_k$, and $E(\sum_{k=0}^{n-1} D_k)$ can be increased by the pairwise exchange of p_k and p_{k+1}; the maximal value of $E(\sum_{k=0}^{n-1} D_k)$ is therefore obtained by the schedule that satisfies $p_1 \geqslant \ldots \geqslant p_n$. This result proves Theorem 1 for Class I.

Class II. Here X_k and D_k are each a mixture of sums of independent exp(1) random variables of which there are 0 or 1 or 2. Recall that if X, Y are distributed as Erlang with parameters k and l, respectively (i.e., as sums of k and l i.i.d. exponential random variables) with rate 1, then $\max(X, Y) - \min(X, Y)$ is a mixture of Erlangs with parameters $1, \ldots, \max(k, l)$. It is easy to verify that, conditional on the values of χ_{k+1} and δ_k we now get for β_{k+1}, the value of $P(\delta_{k+1} = 2)$, conditional on δ_k and χ_{k+1}:

$$\beta_{k+1} = \begin{array}{c|ccc} & & \chi_{k+1} & \\ \delta_k & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1/2 \\ 2 & 1 & 1/2 & 1/2 \end{array} . \qquad (18)$$

We now proceed as for case I; using (12) and (15) which still hold, and setting up a difference equation for β_k, we obtain

$$\beta_k = \left(\frac{1}{2}\right)^k \beta_0 + \sum_{j=1}^{k} (p_j + p_0 \ldots p_j)\left(\frac{1}{2}\right)^{k-j+1}$$

$$k = 1, \ldots, n, \quad (19)$$

$$\mathrm{E}\left(\sum_{k=0}^{n-1} D_k\right)$$

$$= n + 2\left(1 - \left(\frac{1}{2}\right)^n\right)\beta_0 - p_0$$

$$+ \sum_{k=1}^{n-1} p_k - \sum_{k=1}^{n-1} (p_k + p_0 \ldots p_k)\left(\frac{1}{2}\right)^{n-k}. \quad (20)$$

Define Δ as before, as the value of (20) minus the value obtained when p_k, p_{k+1} are exchanged; then

$$\Delta = (1/2)^{n-k}(1 - p_0 \ldots p_{k-1})(p_k - p_{k+1})$$

$$k = 1, \ldots, n-1, \quad (21)$$

and the proof of Theorem 1 for class II follows.

Class III. Let $q_i = 1 - 2p_i$ for $i = 1, \ldots, n$ be the probability that X_i is odd, and let $q_0 = \alpha_0$; we then have

$$\mathrm{E}(D_k \mid D_{k-1} \text{ even}) = 1,$$
$$\mathrm{E}(D_k \mid D_{k-1} \text{ odd}) = 1 - q_k, \quad \text{and}$$
$$\mathrm{E}(D_k) = 1 - \alpha_{k-1}q_k \quad k = 1, \ldots, n. \quad (22)$$

To obtain $\alpha_k = \mathrm{P}(D_k = 1) = \mathrm{P}(D_k \text{ odd})$, we set up the following difference equation:

$$\alpha_k = \alpha_{k-1}(1 - q_k) + (1 - \alpha_{k-1})q_k. \quad (23)$$

We see immediately that

$$\alpha_0 = q_0,$$
$$\alpha_1 = q_0 + q_1 - 2q_0q_1, \text{ and}$$
$$\alpha_2 = q_0 + q_1 + q_2 - 2q_0q_1 - 2q_0q_2 - 2q_1q_2 + 4q_0q_1q_2.$$

Letting $S_r^{(k+1)}$ denote the rth symmetric function of $q_0, \ldots, q_k$, that is,

$$S_r^{(k+1)} = \sum_{0 \le i_1 < \ldots < i_r \le k} \cdots \sum q_{i_1} \ldots q_{i_r} \quad r = 1, \ldots, k+1,$$

we find, by induction, the solution to (23) as

$$\alpha_k = \sum_{r=1}^{k+1} (-2)^{r-1} S_r^{(k+1)}. \quad (24)$$

Substituting in (22) and adding up over $k = 0, \ldots, n-1$, we get

$$\mathrm{E}\left(\sum_{k=1}^{n-1} D_k\right) = (n-1) - \sum_{r=2}^{n} (-2)^{r-2} S_r^{(n)}. \quad (25)$$

We see that $\mathrm{E}(\sum_{k=1}^{n-1} D_k)$ is a symmetric function c $q_0, q_1, \ldots, q_{n-1}$ and any pairwise exchanges, or an permutations of $q_1, \ldots, q_{n-1}$, will not affect (25 Thus the order of the jobs $1, \ldots, n-1$ has no effe on the flowtime. The only decision to be made : which job to do last. We take a derivative with respe to q_{n-1} to get

$$\frac{d}{dq_{n-1}} \mathrm{E}\left(\sum_{k=1}^{n-1} D_k\right) = -\sum_{r=1}^{n-1} (-2)^{r-1} S_r^{(n-1)}$$
$$= -\alpha_{n-2} < 0. \quad (26$$

Therefore, if $p_n > p_{n-1}$, so that $1 - 2p_n = q_n < q_{n-1}$, pairwise exchange between p_{n-1} and p_n will increas $\mathrm{E}(\sum_{k=1}^{n-1} D_k)$. By the symmetry, we can make a simila improvement if $p_n > p_k$ for any $k = 1, \ldots, n -$ Hence the maximum is achieved when p_n : $\min_{1 \le k \le n} p_k$. We have shown that the expected flov time depends only on the last job in the schedule, an is maximized if that job has smallest variance. Th there are many optimal schedules. One of these opt mal schedules is LVF. This result completes the pro of Theorem 1.

Corollary 1. *For $m = 2$ machines, LVF minimiz the expected makespan for classes I, II and III.*

Proof. The makespan $C_{\max}$ is given by

$$C_{\max} = \frac{X_1 + \ldots + X_n + D_n}{2}, \quad (2$$

so we need to minimize $\mathrm{E}(D_n)$. For class I, by (14) ar (15),

$$\mathrm{E}(D_n) = 1 + \left(\frac{2}{3}\right)^n \beta_0 - p_0 \ldots p_n$$
$$+ \frac{1}{3}\sum_{k=1}^{n} (2p_k + p_0 \ldots p_k)\left(\frac{2}{3}\right)^{n-k}. \quad (2$$

For class II, by (15) and (19),

$$\mathrm{E}(D_n) = 1 + \left(\frac{1}{2}\right)^n \beta_0 - p_0 \ldots p_n$$
$$+ \frac{1}{2}\sum_{k=1}^{n} (p_k + p_0 \ldots p_k)\left(\frac{1}{2}\right)^{n-k}. \quad (2$$

It is easy to check by a pairwise exchange of p_k ar p_{k+1} that the corollary holds.

For class III, we note that, as in (25), we have

$$\mathrm{E}\left(\sum_{k=1}^{n} D_k\right) = n - \sum_{r=2}^{n+1} (-2)^{r-2} S_r^{(n+1)}. \quad (3$$

This expression is symmetric in $q_1, \ldots, q_n$, and ther fore independent of the schedule. On the other han

LVF (or any rule that puts $p_n = \min_{1\le k\le n} p_k$), maximizes $E(\sum_{k=1}^{n-1} D_k)$. Hence, LVF minimizes $E(D_n)$, and the corollary is proved.

3. Minimization of Expected Makespan

In this section we prove Theorem 2 for classes I and II for any number of machines m. The proof for class III with $m = 2$ was given in Corollary 1. Assume the m machines are available originally at times $U_0^{(1)} \le \ldots \le U_0^{(m)}$, with $\sum U_0^{(i)} = 0$, and with $U_0^{(i)}$ in class I′ or II′ for $i = 1, \ldots, m$, according to whether the X_j's are in class I or II. Consider jobs $1, \ldots, k$ and let $U_k^{(1)} \le \ldots \le U_k^{(m)}$ be the times at which the machines complete their processing, so that

$$\sum_{i=1}^{m} U_k^{(i)} = X_1 + \ldots + X_k. \tag{31}$$

Define

$$D_k^{(i)} = U_k^{(i+1)} - U_k^{(i)} \quad i = 1, \ldots, m-1, \qquad B_k^{(i)} = U_k^{(m)} - U_k^{(m-i)}. \tag{32}$$

Clearly $C_{\max} = U_n^{(m)}$, and therefore the makespan is

$$C_{\max} = \frac{X_1 + \ldots + X_n + B_n^{(1)} + \ldots B_n^{(m-1)}}{m}$$
$$= X_1 + \ldots + X_n + D_n^{(1)} + 2D_n^{(2)} + \ldots + (m-1)D_n^{(m-1)}. \tag{33}$$

We will show that exchanging p_k and p_{k+1} when $p_k < p_{k+1}$ will lead to all of $D_j^{(1)}, \ldots, D_j^{(m-1)}$ decreasing stochastically for $j = k+1, \ldots, n$, from which the theorem will follow.

Consider again the schedule of jobs $1, \ldots, k$. At time $U_k^{(1)}$ one machine completes a job and is no longer occupied by any of jobs $1, \ldots, k$. The other $m - 1$ machines are still occupied by the (> 0) remainders of some of the jobs $1, \ldots, k$, whose distributions belong to class I′ or II′. Let $\delta_k^{(1)}, \delta_k^{(2)}, \ldots, \delta_k^{(m-1)}$ be defined as before for the $m - 1$ remainder jobs, and define N_k as the number of $\delta_k^{(j)}$'s which are equal to 2, for $k = 0, 1, \ldots, n$.

Proposition. *For jobs of class I or II, let*

$N' = N_{k+1} \mid N_{k-1} = N,\ \chi_k = 2,\ \chi_{k+1} = 1,$ *and*

$N'' = N_{k+1} \mid N_{k-1} = N,\ \chi_k = 1,\ \chi_{k+1} = 2;$

then $N'' \ge_{st} N'$.

Proof. Under these conditioning events, N_{k+1} can assume values $N - 1$, N or $N + 1$. For class I, straightforward calculation (Figure 1) shows

$$\begin{aligned} p(N' = N+1) &= \left(\frac{2m-2N-2}{2m-N-1}\right)^2 \le \frac{2m-2N}{2m-N}\cdot\frac{2m-2N-2}{2m-N-1} \\ &= P(N'' = N+1) \quad \text{and} \\ P(N' = N-1) &= \frac{(N+1)}{2m-N-1}\left(\frac{N}{2m-N}\right) \ge \left(\frac{N}{2m-N}\right)^2 = P(N'' = N-1), \end{aligned} \tag{34}$$

and the proposition follows.

The proof for class II is less direct. Recall that, for class II, if χ_k is 1, X_k is exp(1) while if $\chi_k = 2$, X_k is erlang(2), i.e., a sum of two independent exp(1) random variables. So altogether, for N' and N'', X_k and X_{k+1} consist of 3 independent exp(1) random variables, each of the jobs k and $k + 1$ starts with an exp(1) duration (let X, Y denote these durations), and the third exp(1) random variable (let Z denote its duration) is added, after the initial exp(1) duration to (i) job k in the case of N', and (ii) job $k + 1$ in the case of N''. We now use a sample path argument to show that $N' \le_{ST} N''$. At $U_{k-1}^{(1)}$ we start X together with whatever $m - 1$ remainders are on the other machines. If X is not the first to end, so that another remainder job ends first, we start Y, together with X and $m - 2$ other job remainders; in this case the rest of the schedule has the same stochastic behavior for both N' and N'' and they have the same distribution. If X ends first, we start Z in the realization of N' and Y in the realization of N''. We continue the sample path argument with the assumption that Z in N' equals Y in N''. Following X we now have Z (or Y) running together with $m - 1$ other job remainders. If Z (and Y) end first, the rest of the schedule has the same stochastic behavior for both N' and N'' and they have the same distribution. If, however, another job ends first, then in the realization of N' we start Y immediately in parallel with Z and $m - 2$ other job remainders, while in the realization of N'', Y is running in parallel to the other $m - 2$ job remainders, and Z is waiting for the completion of Y before it can start.

Let $\tilde{N}$ be the number of job remainders out of the $m - 2$ for which $\chi = 2$; then we have $N'' = \tilde{N} + 1$ while $N' \le \tilde{N}$, and the proposition is proved.

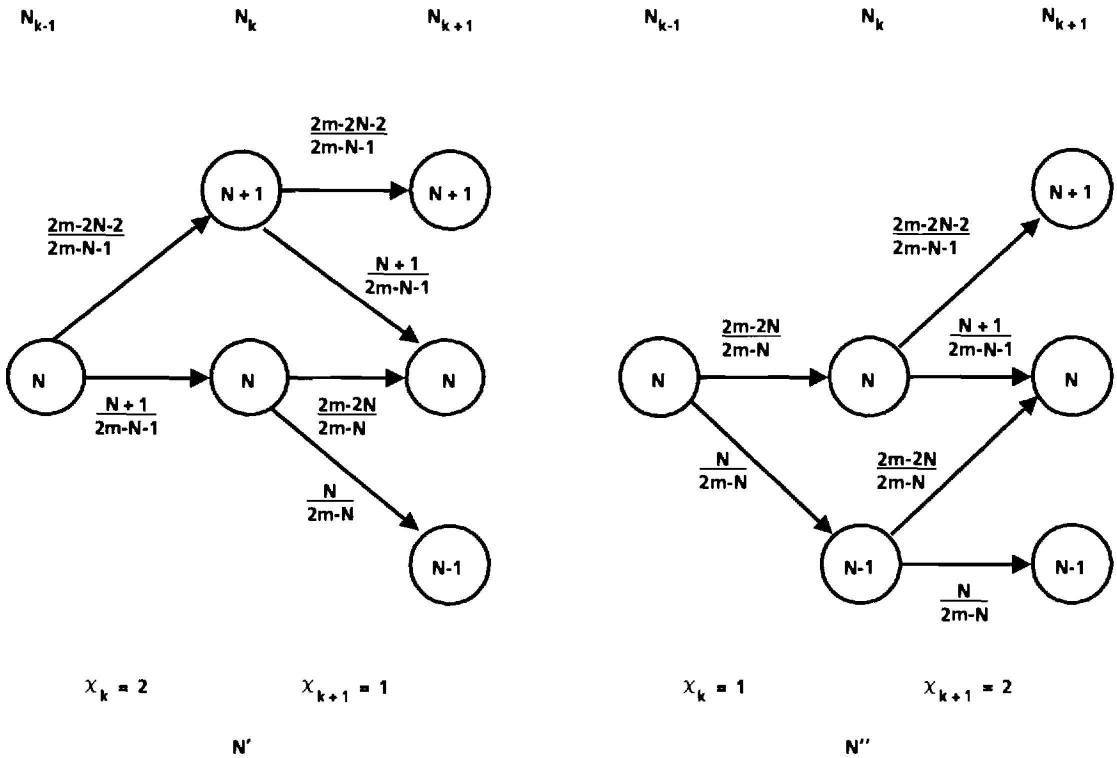


Figure 1. The distribution of N', N''.

This proposition supplies the crucial exchange argument to prove the theorem. Consider changing the order of jobs k and $k + 1$. Doing so will affect the makespan only if one of the jobs has $\chi = 2$ and the other $\chi = 1$. Doing k followed by $k + 1$ we have $\chi_k = 2$, $\chi_{k+1} = 1$, and N_{k+1} distributed like N' with probability $p_k(1 - 2p_{k+1})$; the probability that N_{k+1} is distributed like N'' is analogously $p_{k+1}(1 - 2p_k)$. Reversing the order will add $p_{k+1} - p_k$ to the probability of the first event and decrease the probability of the second event by the same amount. Thus, if $p_{k+1} > p_k$, reversing the order will decrease N_{k+1} stochastically. It is easy to see that $D^{(1)}_{k+1}$ is stochastically increasing in N_{k+1}. So also is N_{k+2}, and so are the numbers of job remainders with $\chi = 2$ when $1, 2, \ldots$ of the $m - 1$ jobs remaining at $U^{(1)}_{k+1}$ are completed. This result proves that the decrease in N_{k+1} will cause $D^{(i)}_j$ for $k + 1 \leqslant j \leqslant n$ and $1 \leqslant i \leqslant m - 1$ to decrease stochastically. We thus complete the proof.

4. Generalizations

In this section we comment on the results of the previous sections and point out some possible extensions.

(a) Preemptive scheduling of class I and II. To minimize expected flowtime, it is clearly optimal to locate and dispose of zero duration jobs first, at no cost. If we were to minimize makespan, zero duration jobs would be irrelevant and could be scheduled anywhere; we can therefore assume we know which are zero duration jobs, and ignore them. The remaining jobs will then have $\chi = 1$ or $\chi = 2$, and will correspond to the model of Weber. The ensuing policies are highest-hazard-rate-first for flowtime, lowest-hazard-rate-first for makespan. These policies differ from the LVF rule.

(b) Dependence of expected flowtime and makespan on $p_1, \ldots p_n$.

Corollary 2. *For every permutation schedule, the expected flowtime is decreasing in each p_j for $j = 1, \ldots, n$ for $m = 2$ machines and jobs of classes, I, II and III. For every permutation schedule the expected makespan is increasing in each p_j for $j = 1, \ldots, n$ for m machines, and for jobs in classes I and II. The expected makespan is also increasing in p_n, where n is the last job, for jobs in class III and for $m = 2$ machines.*

Proof. The proof is conducted by taking derivatives of (16, 20, 26, 28, 29); we omit the details. The dependence of expected makespan on $p_1, \ldots, p_{n-1}$ for jobs in class III with $m = 2$ can also be fully analyzed, but it is not simply monotone.

(c) Generalization of class I and class II. Theorems 2 and 3 can be proved for a family of jobs $X_1, \ldots, X_n$ whenever all the X_j's are a mixture of 0, Y, Z and have the same mean $E(X_j) = \theta$, and either $Y \sim \exp(1)$, $Z \sim \exp(\lambda)$, $0 < \lambda < 1$, or $Y \sim \exp(1)$, $Z \sim \exp(1) * \exp(\lambda)$, $\lambda > 0$.

5. LVF Is Not Optimal for More General 3-Valued Distributions

In this section we examine the case in which the X_j belong to a class of distributions that generalize class III. We find the policy that minimizes both expected flowtime and expected makespan; surprisingly, this policy is not LVF. Similar optimal policies for some other models have been obtained by Pinedo (1981) and Weiss (1984).

We take fixed θ, $-1 < \theta < 1$ and p_j, $q_j = 1 - 2p_j$, $0 < q_j < 1 - |\theta|$ for $j = 1, \ldots, n$ and let

$$X_j = \begin{cases} 0 & \text{w.p.} \quad p_j + \dfrac{\theta}{2} = \dfrac{1 - q_j + \theta}{2}. \\ 1 & \text{w.p.} \quad 1 - 2p_j = q_j. \\ 2 & \text{w.p.} \quad p_j - \dfrac{\theta}{2} = \dfrac{1 - q_j - \theta}{2}. \end{cases} \tag{35}$$

For this class we have

$$E(X_j) = 1 - \theta, \text{ and}$$
$$V(X_j) = 1 - q_j - \theta^2. \tag{36}$$

We note that if $p_k \geqslant p_j$, then $X_k \geqslant_v X_j$, that is, X_k is stochastically more variable than X_j (see Ross 1983, section 8.5 for definition).

We will establish the following theorem.

Theorem 3. *For $m = 2$ machines, and for $X_1, \ldots, X_n$ distributed as (35), all with mean $1 - \theta$, and with $p_1 \geqslant p_2 \geqslant \ldots p_n$, the following rule minimizes expected makespan and the expected flowtime:*

a) *If $\theta > 0$, so that $E(X_j) < 1$, LVF is optimal—that is, the schedule $1, \ldots, n$ is optimal.*
b) *If $\theta = 0$, any policy that has n last is optimal.*
c) *If $\theta < 0$, so that $E(X_j) > 1$, the optimal schedule has the following structure: smallest variance last, largest variance before last, second smallest variance second before last, second largest variance third before last, and so forth—that is, the schedule $\ldots, 2, n-1, 1, n$ is optimal.*

Proof. Let $D_0, D_1, \ldots, D_n$ be defined as in the proof of Theorem 1, with D_k in class III′:

$$D_k = \begin{cases} 0 & \text{w.p.} \quad \gamma_k + \dfrac{\eta_k}{2} = \dfrac{1 - \alpha_k + \eta_k}{2}, \\ 1 & \text{w.p.} \quad 1 - 2\gamma_k = \alpha_k, \\ 2 & \text{w.p.} \quad \gamma_k - \dfrac{\eta_k}{2} = \dfrac{1 - \alpha_k - \eta_k}{2}, \end{cases} \tag{37}$$

with initial values for D_0 given by $q_0 = \alpha_0$ and η_0, so that

$$E(D_k) = 1 - \eta_k, \quad \text{and}$$
$$V(D_k) = 1 - \alpha_k - \eta_k^2. \tag{38}$$

The following difference equations can be set up for α_k and η_k:

$$\alpha_k = \alpha_{k-1} + q_k - 2\alpha_{k-1}q_k, \quad \text{and} \tag{39}$$

$$\eta_k = \alpha_{k-1}q_k + \theta\eta_{k-1}. \tag{40}$$

Equation 39 is the same as (23), and has (24) as its explicit solution, in terms of $q_0, \ldots, q_n$; Equation 40 is solved by

$$\eta_k = \theta^k\eta_0 + \theta^{k-1}\alpha_0 q_1 + \theta^{k-2}\alpha_1 q_2 + \ldots + \alpha_{k-1}q_k, \tag{41}$$

and we get

$$E(D_n) = 1 - \theta^n\eta_0 - \sum_{j=1}^{n} \theta^{n-j}\alpha_{j-1}q_j, \tag{42}$$

$$E\left(\sum_{j=0}^{n-1} D_j\right) = n - \frac{1-\theta^n}{1-\theta}\eta_0 - \sum_{j=1}^{n-1} (1 - \theta^{n-j})\alpha_{j-1}q_j. \tag{43}$$

Recall that to minimize expected makespan we need to minimize $E(D_n)$, and to minimize expected flowtime we need to maximize $E(\sum_{j=0}^{n-1} D_j)$.

Let $\Delta_F(k, l)$ be the value of $E(\sum_{j=0}^{n-1} D_j)$ minus the value with the order of jobs k and l exchanged, and let $\Delta_M(k, l)$ be similarly defined for $E(D_n)$. A schedule for which $\Delta_F(k, l) < 0$ ($\Delta_M(k, l) > 0$) for some pair k and l cannot be optimal for flowtime (makespan). Hence, $\Delta_F(k, l) \geqslant 0$ ($\Delta_M(k, l) \leqslant 0$) for all k and m is a necessary condition for minimal expected flowtime (makespan). Using (42, 43) we obtain the following results:

$$\Delta_F(k, k+1) = \theta^{n-k-1}(q_{k+1} - q_k)\alpha_{k-1} \qquad k = 1, \ldots, n-1, \tag{44}$$

$$\Delta_F(k-1, k+1) = \theta^{n-k-1}(q_{k+1} - q_{k-1})(\alpha_{k-2}(1 - q_k + \theta) + q_k(1 - \alpha_{k-2})) \quad k = 2, \ldots, n-1, \tag{45}$$

and

$$\Delta_M(k, l) = -(1 - \theta)\Delta_F(k, l). \tag{46}$$

For $\theta > 0$, (44) implies that $q_n \geqslant q_{n-1} \geqslant \ldots \geqslant q_1$, is necessary for flowtime optimality, and, since this condition uniquely determines the schedule, and an optimum exists, it is sufficient; this result proves (a) for flowtime.

For $\theta < 0$, (44) and (45) imply the following optimal orders:

For n even, $q_n \geqslant q_{n-2} \geqslant \ldots \geqslant q_2 \geqslant q_1 \geqslant q_3 \geqslant \ldots q_{n-3} \geqslant q_{n-1}$.

For n odd, $q_n \geqslant q_{n-2} \ldots \geqslant q_3 \geqslant q_1 \geqslant q_2 \geqslant \ldots q_{n-3} \geqslant q_{n-1}$.

This conclusion proves (c) for flowtime.

The proof of (a) and (c) for makespan follows from (46). (b) was proved in Section 2.

Acknowledgment

This work was supported in part by National Science Foundation grants ECS-83-1223 (Professor Pinedo) and ECS-87-12798 (Professor Weiss).

References

BRUNO, J., P. DOWNEY AND G. FREDERICKSON. 1981. Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan. *J. Assoc. Comput. Mach.* **28,** 100–113.

GLAZEBROOK, K. D. 1979. Scheduling Tasks with Exponential Service Times on Parallel Processors. *J. Appl. Prob.* **16,** 685–689.

PINEDO, M. 1981. Minimizing Makespan with Bimodal Processing Time Distributions. *Mgmt. Sci.* **27,** 582–587.

PINEDO, M., AND G. WEISS. 1979. Scheduling Stochastic Tasks on Two Parallel Processors, *Naval Res. Logist. Quart.* **26,** 527–536.

PINEDO, M., AND S. H. WIE. 1984. Inequalities for Stochastic Flow Shops and Stochastic Job Shops. (submitted to *Naval Research Logistics Quarterly*).

ROSS, S. M. 1983. *Stochastic Processes.* John Wiley & Sons, New York.

WEBER, R. R. 1982. Scheduling Jobs with Stochastic Processing Requirements on Parallel Machines to Minimize Makespan or Flow Time. *J. Appl. Prob.* **19,** 167–182.

WEBER, R. R., P. VARAIYA AND J. WALRAND. 1986. Scheduling Jobs with Stochastically Ordered Processing Times on Parallel Machines to Minimize Expected Flowtime. *J. Appl. Prob.* **23,** 841–847.

WEISS, G. 1984. Scheduling Spares with Exponential Lifetimes in a Two-Component Parallel System. *Naval Res. Logist. Quart.* **31,** 431–446.

WEISS, G., AND M. PINEDO. 1980. Scheduling Tasks with Exponential Service Times on Nonidentical Processors to Minimize Various Cost Functions. *J. Appl. Prob.* **17,** 187–202.

Industrial and Systems Engineering
Report Series No. J-87-12
November, 1987

A WORST CASE ANALYSIS OF SMITH'S RULE FOR SCHEDULING PARALLEL MACHINES TO MINIMIZE WEIGHTED FLOWTIME

Gideon Weiss*, Chueng-Chiu Huang*, Chung-Lun Li**,
Shucheng Liu**, Michael Pinedo**, Jingsheng Song**.

* Research supported in part by NSF grant ECS-8712798.

* School of Industrial and Systems Engineering, Georgia Institute of Technology

** Department of Industrial Engineering and Operations Research, Columbia University.

A WORST CASE ANALYSIS OF SMITH'S RULE FOR SCHEDULING PARALLEL MACHINES TO MINIMIZE WEIGHTED FLOWTIME

Gideon Weiss*, Chueng-Chiu Huang*, Chung-Lun Li**,
Shucheng Liu**, Michael Pinedo**, Jingsheng Song**.

November 1987.

ABSTRACT

Minimization of the weighted flowtime (weighted sum of completion times) of a batch of jobs on several parallel machines is NP-hard. Smith's rule is to start the jobs in decreasing order of weight to processing time ratio. We show: The weighted flowtime under Smith's rule can approach but not exceed $1/2 + \sqrt{1/2}$ times the optimal value.

Keywords: scheduling, parallel machines, weighted flowtime, worst case performance.

* Research supported in part by NSF grant ECS-8712798.

* School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

** Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027.

INTRODUCTION AND RESULT

A batch of n jobs is to be scheduled on a set of $M+1$ parallel machines. Job j requires processing time p_j and has weight (holding cost per unit time) w_j, $j=1,\cdots,n$. For any given schedule let c_j be the completion time of job j; then $\Sigma_{j=1}^{n} w_j c_j$ is the weighted flowtime of the schedule; see Conway Maxwell and Miller [2]. The objective is to find a schedule to minimize it.

McNaughton [8] shows that the optimal schedule is work conserving, that is no machine is left idle while any jobs remain to be started. Hence, the problem is to chose a permutation which will give the optimal order in which to start the jobs. This problem is unary NP-hard for unrestricted M, binary NP-hard for fixed $M \geq 1$ (Problem SS13 in Garey and Johnson [5], Lenstra, Rinnooy-Kan and Brucker [7]). Sanhi [9] provides an $O[n(n^2 2^L)^{M+1}]$ pseudo polynomial time algorithm for the problem. Eastman, Even and Isaacs [3] derive some inequalities which can be used for a branch and bound scheme; see also Elmaghraby and Park [4], Barnes and Brennan [1].

Smith's [10] rule, which is optimal for a single machine, is to start the jobs in decreasing order of w_j/p_j. We want to investigate its performance. In this we follow some conjectures and earlier attempts of Huang, Li and Pinedo [6].

Weiss [11] has recently looked at the stochastic version of the problem, when the processing times p_j are unknown, and are drawn from some probability destributions F_j, $j=1,\cdots,n$. Under an assumption that the distributions (but not the processing times themselves) are uniformly bounded in some sense, he shows that the <u>difference</u> between the ecpected weighted flowtime using Smith's rule (with $E(p_j)$ replacing p_j), and using an optimal schedule, is bounded by a constant. Weiss also conjectures that Smith's rule has a turnpike property - it actually provides the optimal decision at most times. this suggests that Smith's rule is an almost ideal heuristic for the stochastic problem.

In the deterministic case an assumption that all the processing times are uniformly bounded is untenable. In its absence the difference between weighted flowtime under SR (Smith's rule) and under OPT (Optimal schedule) is unbounded. The ratio is however bounded as we show:

<u>Theorem 1</u>: The ratio between $\Sigma_{j=1}^{n} w_j c_j$ under SR and under OPT may approach but not exceed $1/2 + \sqrt{1/2} = 1.2071\ldots$

$$R = \sup \left(\frac{\Sigma_{j=1}^{n} w_j c_j \mid SR}{\Sigma_{j=1}^{n} w_j c_j \mid OPT} \,\middle|\, M, n, p_1, \cdots, p_n, w_1, \cdots, w_n \right) = \frac{1}{2} + \sqrt{\frac{1}{2}} \quad (1)$$

PROOF OF THEOREM

The proof follows from a series of propositions.

Proposition 1: R is approached when w_j is nearly equal p_j $j=1,\cdots,n$.

$$R = \sup \left(\frac{\Sigma_{j=1}^{n} p_j c_j \mid SR}{\Sigma_{j=1}^{n} p_j c_j \mid OPT} \;\middle|\; M, n, p_1, \cdots, p_n, w_1 \cong p_1, \cdots, w_n \cong p_n \right).$$

Proof: Assume jobs are ordered so that $w_1/p_1 \geq \cdots \geq w_n/p_n$. For any schedule write:

$$\sum_{j=1}^{n} w_j c_j = \frac{w_n}{p_n} \sum_{j=1}^{n} p_j c_j + \sum_{k=1}^{n-1} \left(\frac{w_k}{p_k} - \frac{w_{k+1}}{p_{k+1}} \right) \sum_{j=1}^{k} p_j c_j . \tag{2}$$

By (2) the ratio for $\Sigma_{j=1}^{n} w_j c_j$ is a weighted average of the ratios for $\Sigma_{j=1}^{k} p_j c_j$ $(k=1,\cdots,n)$, under the original schedules SR and OPT. Note that SR for the jobs $1,\cdots,n$ with the weights $w_1,\cdots,w_n$ induces the SR schedule for the set of jobs $1,\cdots,k$ with the weights $p_1,\cdots,p_k$ (in contrast, OPT will typically not be optimal for the k job problem, and it may insert idle times and not be work conserving). The proposition follows. □

Proposition 2: Assume from here onwards that $w_j = p_j$, $j=1,\cdots,n$. Then:

$$R = \sup \left(\frac{\max \left(\Sigma_{j=1}^{n} p_j c_j \mid \text{all schedules} \right)}{\min \left(\Sigma_{j=1}^{n} p_j c_j \mid \text{all schedules} \right)} \;\middle|\; M, n, p_1, \cdots, p_n \right) \tag{3}$$

Proof: Slight perturbations can induce $w_j \cong p_j$ such that the SR order is one which maximizes $\Sigma_{j=1}^{n} p_j c_j$. □

Proposition 3: Let $u_0 \leq \cdots \leq u_M$ be the times at which the $M+1$ machines finish all processing; let $d_i = u_i - u_{i-1}$, $i=1,\cdots,M$; let

$$S_n^2 = \frac{1}{M} \sum_{i=0}^{M} u_i^2 - \frac{1}{M(M+1)} \left(\sum_{i=0}^{M} u_i \right)^2$$

$$= \frac{1}{M} \sum_{i=1}^{M} d_i^2 - \frac{1}{M(M+1)} \left(\sum_{i=1}^{M} d_i \right)^2$$

Then:

$$\sum_{j=1}^{n} p_j c_j = \frac{1}{2}\sum_{j=1}^{n} p_j^{\,2} + \frac{1}{2}\sum_{i=0}^{M} u_i^{\,2} \tag{4a}$$

$$= \frac{1}{2(M+1)}\left(\sum_{j=1}^{n} p_j\right)^2 + \frac{1}{2}\sum_{j=1}^{n} p_j^{\,2} + \frac{M}{2} S_n^{\,2}. \tag{4b}$$

This holds for all schedules, including schedules which are not work conserving, as long as all the jobs on each machine are processed with no idle time.

Proof: Consider scheduling jobs $1,\cdots,k$ on a single machine in that order. Then:

$$\sum_{j=1}^{k} p_j c_j = \sum_{j=1}^{k} p_j(p_1 + \cdots + p_j) = \frac{1}{2}\left(\sum_{j=1}^{k} p_j\right)^2 + \frac{1}{2}\sum_{j=1}^{k} p_j^{\,2}. \tag{5}$$

This is independent of the order of the jobs and so holds for all schedules. Let $j_1,\cdots,j_i$ be the jobs done on machine i, apply (5) and sum over $i=0,\cdots,M$ to get (4a). (4b) is obtained by substitution, since $\Sigma_{j=1}^{n} p_j = \Sigma_{i=0}^{M} u_i$. That the two definitions of $S_n^{\,2}$ are equivalent is well known. □

Proposition 4: R is approached when n is large and when all but $p_1,\cdots,p_M$ are infinitesimaly small so that $\Sigma_{j=M+1}^{n} p_j = M+1$, while $\Sigma_{j=M+1}^{n} p_j^{\,2} \cong 0$.

Proof: Substitute (4b) into (3) to get:

$$R = \sup\left(\frac{\frac{1}{2(M+1)}\left(\Sigma_{j=1}^{n} p_j\right)^2 + \frac{1}{2}\Sigma_{j=1}^{n} p_j^{\,2} + \frac{M}{2}\bar{S}_n^{\,2}}{\frac{1}{2(M+1)}\left(\Sigma_{j=1}^{n} p_j\right)^2 + \frac{1}{2}\Sigma_{j=1}^{n} p_j^{\,2} + \frac{M}{2}\underline{S}_n^{\,2}} \;\middle|\; M,n,p_1,\cdots,p_n \right) \tag{6}$$

where $\bar{S}_n^{\,2}$ and $\underline{S}_n^{\,2}$ are the max and min over all schedules. Rename jobs $1,\cdots,M$ to be the last jobs to finish in the numerator. Replace each job j, $j>M$, by many short jobs whose processing times add up to p_j, and consider the new problem with all the chopped up jobs scheduled in the same periods as the original jobs were. The only change in (6) will then be a reduction in $\Sigma_{j=1}^{n} p_j^2$, which will increase the ratio. Maximizing the numerator and minimizing the denominator over all possible schedules of the new problem may increase the ratio even further. The choice $\Sigma_{j=M+1}^{n} p_j = M+1$ is done by rescaling. □

Proposition 5: Let $p_1 \leq \cdots \leq p_M$ and define y and k by:

$$p_k \leq y = \frac{M+1 + \Sigma_{j=1}^{k} p_j}{k+1} \leq p_{k+1}, \tag{7}$$

(denote $p_0 = 0$, $p_{M+1} = \infty$ to allow for $k=0$, $k=M$; k is not unique if one or both of the inequalities hold as an equality, in such a case the stated result will hold for all k satisfying (7)). Then:

$$R = \max\left(\frac{\frac{M+1}{2} + \sum_{j=1}^{M} p_j + \sum_{j=1}^{M} p_j^2}{\frac{1}{2(k+1)} (M+1 + \sum_{j=1}^{k} p_j)^2 + \frac{1}{2} \sum_{j=1}^{k} p_j^2 + \sum_{j=k+1}^{M} p_j^2} \middle| M,k,p_1,\cdots,p_M \right) \tag{8}$$

Proof: For the numerator, reconsider (6). If any of jobs $1,\cdots,M$ start prior to u_0, the part performed before u_0 can be chopped up as in proposition 4, and a problem with a higher ratio is obtained. Hence, the only problems that need to be considered are problems for which $\Sigma_{j=1}^{n} p_j c_j$ is maximized when all the jobs $1,\cdots,M$ start at $u_0 = 1$, after completion of all the infinitesimal jobs. The numerator for such problems is, by (4a):

$$\sum_{j=1}^{n} p_j c_j = \frac{1}{2} \sum_{j=1}^{M} p_j^2 + \frac{1}{2} (1 + \sum_{j=1}^{M} (p_j+1)^2) \tag{9}$$

For the denominator, by work conservation of the optimal schedule, we can assume that jobs $1,\cdots,M$ are done on machines $1,\cdots,M$ each job on a different machine, and that machine 0 which processes only infinitesimal jobs, is the first machine to finish all processing at u_0. If any job $1 \le j \le M$ does not start at 0 and is completed after u_0, increasing u_0 and starting job j earlier reduces S_n^2, and if the resulting schedule is not work conserving, further reduction of $\Sigma_{j=1}^{n} p_j c_j$ is possible. Hence, all jobs $j=1,\cdots,M$ either start at 0 or complete before u_0. It follows that $u_i=y$ for $i=0,\cdots,k$, and, by (4a):

$$\sum_{j=1}^{n} p_j c_j = \frac{1}{2}(k+1)\, y^2 + \frac{1}{2}\sum_{j=1}^{k} p_j^2 + \sum_{j=k+1}^{M} p_j^2 \qquad (10)$$

Expanding the ratio of (9) and (10) yields (8). □

<u>Propposition 6</u>: R is achieved when $p_{k+1} = \cdots = p_M = X$, so that:

$$R = \max\left(\frac{\frac{M+1}{2} + \sum_{j=1}^{k} p_j + \sum_{j=1}^{k} p_j^2 + (M-k)(X^2+X)}{\frac{M+1}{k+1}\left(\frac{M+1}{2} + \sum_{j=1}^{k} p_j\right) + \frac{1}{2(k+1)}\left(\sum_{j=1}^{k} p_j\right)^2 + \frac{1}{2}\sum_{j=1}^{k} p_j^2 + (M-k)X^2} \;\middle|\; M,k,p_1,\cdots,p_k,X \right) \qquad (11)$$

<u>Proof</u>: Consider (8). $p_{k+1},\cdots,p_M$ appear as $\Sigma_{j=k+1}^{M} p_j + \Sigma_{j=k+1}^{M} p_j^2$ in the numerator, and as $\Sigma_{j=k+1}^{M} p_j^2$ in the denominator. Keeping the sum of squares constant, R is maximized when the sum is maximized, which happens when all the p_j are equal. The value is then $X = \sqrt{\frac{1}{M-k} \Sigma_{j=k+1}^{M} p_j^2} \ge p_{k+1} \ge y$. □

<u>Proposition 7</u>: R is achieved when each of p_j $j=1,\cdots,k$ is at either the lower or the upper end of its range.

<u>Proof</u>: Taking the derivative of the ratio in (11) with respect to p_ℓ, one finds that this derivative has the same sign as a convex parabola in p_ℓ, with a negative free term. Hence, the ratio is initially decreasing for $p_\ell \ge 0$, reaches a minimum, and is increasing thereafter. □

Proposition 8: R is achieved as

$$R = \max \left(\frac{\frac{M+1}{2} + (M-k)(X^2+X)}{\frac{(M+1)^2}{2(k+1)} + (M-k)X^2} \;\middle|\; M,\ 0 \le k \le M,\ X \ge \frac{M+1}{k+1} \right) \tag{12}$$

Proof: Assume we have $X > y > p_k \ge \cdots \ge p_1$. If $y > p_\ell > 0$, by proposition 7, the ratio will increase if p_ℓ is pushed to one end of its range. Four possibilities arise: (a) p_ℓ is reduced to 0. (b) p_ℓ is reduced until y hits the value p_k. (c) p_ℓ is increased to y. (d) p_ℓ is increased until y hits X. If (b) or (c) occur, reindex jobs to have $p_1 \le \cdots \le p_k = y$, then reduce k until again $y > p_k$, then use proposition 6 to change $p_{k+1},\cdots,p_M$ to a new common value $X > y$. If (a), (b) or (c) occur a modified schedule with $p_\ell=0$ or with smaller k is obtained. Repeating for other $y > p_j > 0$, leads to a schedule with $p_1= \cdots = p_k= 0$, or to case (d). In case (d), $k = M$, and $X = y = \frac{M+1 + \Sigma_{j=1}^{M} p_j}{M+1} \ge p_M \ge \cdots \ge p_1$. It is easy to see that for fixed M and X under these conditions, the ratio is maximized when $\frac{M+1}{k} \ge X > \frac{M+1}{k+1}$, $0 = p_1 = \cdots = p_{k-1} < p_k \le p_{k+1}= \cdots = p_M = X$ and $p_k = (k+1)X - (M+1)$. It is possible to show that $p_k=0$ or $p_k=y$ is the only case that one needs to consider. However, instead of doing so it is shown in the next proposition that the ratio in case (d) is always less than the conjectured value of R. (12) follows. □

Proposition 9: Assume $0 = p_1= \cdots = p_{k-1}\le p_k\le p_{k+1}= \cdots = p_M= X$ and $y = X$. Then the supremum of the ratio is $4\,(1 - \sqrt{1/2}) = 1.1716...$

Proof: Assume first that $p_k=0$, and therefore $X = \frac{M+1}{k+1}$. Let $\tilde{R}$ be the ratio as in (11). Since in the denominator $u_0= \cdots = u_M= X$, one can recalculate it using (4a).

$$\tilde{R} = \max\left(\frac{\frac{M+1}{2} + (M-k)\ (X+X^2)}{\frac{1}{2}(M-k)\ X^2 + \frac{1}{2}(M+1)\ X^2} \;\middle|\; M,\ k,\ X = \frac{M+1}{k+1} \right)$$

Substituting the value of X and then denoting $\alpha = \frac{k+1}{M+1}$ we get

$$\tilde{R} = \max\left(\frac{\frac{1}{2} + (1 - \frac{k+1}{M+1})\ (\frac{M+1}{k+1} + (\frac{M+1}{k+1})^2)}{(1 - \frac{1}{2}\frac{k+1}{M+1})\ (\frac{M+1}{k+1})^2} \;\middle|\; M,\ k \right)$$

$$= \max\left(\frac{2 - \alpha^2}{2 - \alpha} \;\middle|\; M,\ k,\ \alpha = \frac{k+1}{M+1} \right) \le \max\left(\frac{2 - \alpha^2}{2 - \alpha} \;\middle|\; 0 \le \alpha \le 1 \right)$$

$$= 4\ (1 - \sqrt{1/2}).$$

Let now $\delta = p_k/X$, $0 \le \delta \le 1$, then $X = \frac{M+1}{k+1-\delta}$, and

$$\tilde{R} = \max\left(\frac{\frac{M+1}{2} + (M-k)\ (X+X^2) + \delta X + \delta^2 X^2}{\frac{1}{2}(M-k)\ X^2 + \frac{1}{2}(M+1)\ X^2 + \frac{1}{2}\delta^2 X^2} \;\middle|\; M,\ k,\ 0 \le \delta \le 1,\ X = \frac{M+1}{k+1-\delta} \right)$$

substituting the value of X,

$$\tilde{R} = \max\left(\frac{\frac{1}{2} + (1 - \frac{k+1-\delta}{M+1})\ (\frac{M+1}{k+1-\delta} + (\frac{M+1}{k+1-\delta})^2) - \delta(1-\delta)X^2}{(1 - \frac{1}{2}\frac{k+1-\delta}{M+1})\ (\frac{M+1}{k+1-\delta})^2 - \frac{1}{2}\delta(1-\delta)X^2} \;\middle|\; M,\ k,\ 0 \le \delta \le 1 \right)$$

using the value $\alpha = \frac{k+1-\delta}{M+1}$ and noting that deletion of the last term in the numerator as well as in the denominator increases the ratio, this is

$$\le \max\left(\frac{2 - \alpha^2}{2 - \alpha} \;\middle|\; 0 \le \alpha \le 1 \right)$$

and the proposition follows □

<u>Proposition 10</u>: For fixed M and k, let $\alpha = \frac{k+1}{M+1}$. Then the ratio is maximized by $X = \frac{1}{2\alpha}\left(1 + \sqrt{\frac{1+\alpha}{1-\alpha}}\right)$ and

$$R = \max\left(1 + \frac{1}{2}\left(\sqrt{1-\alpha^2} - (1-\alpha)\right) \;\middle|\; M,\ k,\ \alpha = \frac{k+1}{M+1} \right) \tag{13}$$

<u>Proof</u>: Rewrite R in (12) as

$$R = \max\left(1 + \frac{X - \frac{1}{2\alpha}}{X^2 + \frac{1}{2\alpha(1-\alpha)}} \right)$$

and (13) follows by calculus. □

Proposition 11: $R = 1/2 + \sqrt{1/2}$.

Proof: Direct calculus from (13). □

To summarize: The supremum $R = 1/2 + \sqrt{1/2}$ is approached when $w_j \cong p_j$, there is a large number of infinitesimaly small jobs adding up to M+1 and with a negligible sum of squares, and there are M-k jobs with equal processing times, $X = \frac{1}{2\alpha}(1 + \sqrt{\frac{1+\alpha}{1-\alpha}})$, $\alpha = \frac{k+1}{M+1}$, and, k and M are such that $\alpha \cong \sqrt{1/2}$.

REFERENCES

[1] Barnes J.W., Brennan J.J. (1977) An improved algorithm for scheduling jobs on identical machines. *AIIE Trans.* 9, 25-31.

[2] Conway R.W., Maxwell W.L., Miller L.W. (1967) *Theory of scheduling.* Addison Wesley, Reading MA.

[3] Eastman W.L., Even S., Isaacs I.M. (1964) Bounds for the optimal scheduling of n jobs on m processors. *management Science* 11, 268-279.

[4] Elmaghraby S.E., Park S.H. (1974) Scheduling jobs on a number of identical machines. *AIIE Trans.* 6, 1-12.

[5] Garey M.R., Johnson D.S. (1979) *Computers and Intractability: a Guide to the Theory of NP Completeness.* Freeman, San Francisco.

[6] Huang C.C., Li C.L., Pinedo M. (1987) Worst case analysis of the weighted completion time on two parallel machines. Technical Report, Dept of IEOR, Columbia University.

[7] Lenstra J.K., Rinnooy-Kan A.H.G., Brucker P. (1977) Complexity of machine scheduling problems. *Ann. Discrete Math.* 1, 343-362.

[8] McNaughton R. (1959) Scheduling with deadlines and loss functions. *Management Science* 6, 1-12

[9] Sanhi S. (1976) Algorithms for scheduling independent tasks. *J. Assoc. Comput. Mach.* 23, 116-127.

[10] Smith W.E. (1956) Various optimizers for single stage prroduction. *Naval Res. Logist. Quart.* 3, 59-66.

[11] Weiss G. (1987) Approximation results in parallel machines stochastic scheduling. In preparation.

THE "LARGEST VARIANCE FIRST" POLICY IN SOME STOCHASTIC SCHEDULING PROBLEMS

MICHAEL PINEDO
Columbia University, New York, New York

GIDEON WEISS
Georgia Institute of Technology, Atlanta, Georgia

(Received June 1985; revision received August 1986; accepted February 1987)

We consider a situation in which n jobs, requiring random amounts of processing, all with the same mean, are to be scheduled on m parallel machines with respect to one of two objectives: expected flowtime and expected makespan. We discuss optimality of the rule that says to schedule the jobs with the largest variance first (LVF). We show that for some very simple job length distributions, LVF minimizes both the expected flowtime and the expected makespan.

n trying to solve scheduling problems with jobs whose processing times are not known precisely ı advance but are drawn from given probability stributions, researchers have concentrated on rules ısed on the jobs' mean processing times. One natural ›proach is to use the expected values as stochastic ırrogates for the exactly known processing times of :terministic jobs. Yet an essential role in stochastic heduling is played by the variability that such jobs troduce into the schedule: this problem feature is :tter described by the variances than by the means the jobs. In this paper, we investigate scheduling les based on the variance of the distributions from hich the processing times of the jobs are drawn.

Jobs $1, \ldots, n$ are to be processed without preıptions on m parallel identical machines; job j reıires processing for a duration X_j on one machine; $, \ldots, X_n$ are independent random variables. Jobs ll be completed at time $C_1, \ldots, C_n$, which depend ı the X_j's as well as on the scheduling rule. Two mmonly used objective functions for this model are ɛ flowtime $\sum C_j$ and the makespan $C_{\max} = \max C_j$, d the resulting problem is to find schedules that inimize the expected value of either function. As is own, the longest-expected-processing-time-first apɔach (LEPT) minimizes expected $C_{\max}$ for exponenl jobs or for remainders of independent, identically stributed (i.i.d.) decreasing hazard rate jobs (Bruno, ›wney and Frederickson 1981, Pinedo and Weiss 80, and Weber 1982. In general, LEPT is a good t not optimal heuristic. Recently, in a remarkable per, Weber, Varaiya and Walrand (1986) showed that the shortest-expected-processing-time-first approach (SEPT) minimizes expected flowtime, $\sum C_j$, under the very general condition that the job processing times are stochastically comparable (stochastic comparison is defined by $X \leqslant_{ST} Y$ if $P(X > x) \leqslant P(Y > x)$ for all x); this result generalizes previous work on the problem by Glazebrook (1979), Bruno, Downey and Frederickson, Weiss and Pinedo (1980) and Weber.

In this paper we will look at the case in which all jobs have equal mean processing times, $E(X_1) = \ldots = E(X_n) = 1$. In this case SEPT and LEPT are of course meaningless, and we need to look for other features to compare jobs and to improve our intuitive understanding of the problem.

For $E(X_j) = 1$ for $j = 1, \ldots, n$ we note, first, that, if we have a single machine, then $E(\sum C_j) = n(n+1)/2$ and $E(C_{\max}) = n$, independent of the schedule. Second, processing times are not stochastically comparable unless they are identically distributed. With equal means we naturally turn to the variances, and we find heuristic indications that the rule of LVF, largest variance first, provides good schedules for both objectives—expected flowtime and expected makespan.

The reasoning is as follows: In minimizing makespan, we are concerned about the possibility of being stuck with some long jobs at the end of the process, when some machines are idle and our total processing rate is lower than m; this situation seems to be more likely when the more variable jobs are left to be done last, so LVF is attractive. To minimize flowtime, our

ɔject classification: 592 priority scheduling by variance of stochastic processing time.

0030-364X/87/3506-0884 $01.25

main concern in scheduling earlier jobs is to prevent delays to later jobs, of which there are a large number at the start of the schedule. If we schedule the m jobs with the largest variances first, we are likely to have the first job completion as early as possible, so LVF is attractive.

We do not know how widely applicable this heuristic is and in what situations it might supply the optimal policy. In this paper we examine some particularly simple special cases for which LVF is indeed optimal for makespan as well as flowtime. We chose these special cases because they lend themselves to easy analysis, rather than because of their practical importance. Nevertheless, we feel that they provide a strong case for the use of a LVF heuristic when the mean processing times are equal or nearly equal. (Pinedo and Wie 1984 discuss the performance of LVF for machines in series.)

We consider three families of distributions.

Class I

$$F_j(x) = 1 - (1 - 2p_j)e^{-x} - p_j e^{-x/2} \qquad j = 1, \ldots, n. \quad (1)$$

Class II

$$F_j(x) = 1 - (1 - p_j)e^{-x} - p_j x e^{-x} \qquad j = 1, \ldots, n. \quad (2)$$

Class III

$$\begin{aligned} P(X_j = 0) &= p_j \\ P(X_j = 1) &= 1 - 2p_j \\ P(X_j = 2) &= p_j \end{aligned} \qquad j = 1, \ldots, n. \quad (3)$$

In Sections 2 and 3, we will prove the following theorems.

Theorem 1. *Assume $X_1, \ldots, X_n$ belong to one of the classes, I, II, or III. For $m = 2$ machines, LVF minimizes expected flowtime.*

Theorem 2. *Assume $X_1, \ldots, X_n$ belong to one of the classes I or II, with any number of machines m, or to class III with $m = 2$ machines. Then LVF minimizes expected makespan.*

Each distribution in class I, II, and III is a mixture of 0 and the random variables Y and Z, in the proportions p_j, $1 - 2p_j$ and p_j, if we assume that $E(Y) = 1$, $E(Z) = 2$ and for the three classes we have the following situations.

Class I: $Y \sim \exp(1)$, $Z \sim \exp(½)$.

Class II: $Y \sim \exp(1)$, $Z \sim \exp(1) * \exp(1)$(i.e., convolution of two exponentials with mean 1).

Class III: $Y = 1$, $Z = 2$.

Thus $E(X_j) = 1$ for all j, and the variance $V(X_j)$ i increasing in p_j, specifically

$$V(X_j) = \begin{cases} 1 + 4p_j & \text{class I} \\ 1 + 2p_j & \text{class II} \\ 2p_j & \text{class III.} \end{cases} \quad (4$$

We shall refer to mixtures of 0, Y, Z in the genera proportions $1 - \alpha - \beta \geq 0$, $\alpha \geq 0$, $\beta \geq 0$ as class I′ II′, III′.

In Section 4 we discuss three topics: preemptiv scheduling of jobs of classes I–III, monotonicity o $E(\sum C_j)$ and $E(C_{\max})$ with the parameters p_j, an robustness of the results under wider classes o distributions. We indicate how the optimality o LVF extends to generalizations of both class I an II distributions. On the other hand, LVF is no always optimal: in Section 5 we consider a generali zation of class III for which it is sometimes optimal t alternate jobs with large and with small variances i the schedule.

2. Minimization of Expected Flowtime

In this section we prove Theorem 1. We also obtain as Corollary 1, Theorem 2 for the case $m = 2$. Ther are $m = 2$ machines that become available at time $-D_0/2$, $D_0/2$. The jobs $1, \ldots, n$ have processing time $X_1, \ldots, X_n$, all of which belong to one of the classe I, II, III, with mixing parameters $p_1, \ldots, p_n$. W assume that D_0 is a random variable belonging to th corresponding class I′, II′, III′. The jobs are started i the order $1, \ldots, n$, with completion times $C_1, \ldots$ C_n. We shall see how the expected flowtime $E(\sum C_k$ depends on $p_1, \ldots, p_n$ and show that of all possibl permutations of the values of $p_1, \ldots, p_n$ the one wit $p_1 \geq p_2 \ldots \geq p_n$ minimizes the expected flowtime This result will prove the optimality of LVF amon all permutation rules, and by using induction and th initial value, D_0, will prove the optimality of LV among all the nonpreemptive scheduling policies.

Consider the schedule of jobs $1, \ldots, k$ and le $U_k \leq V_k$ be the times at which the two machines finis to process all these jobs. Also, let $D_k = V_k - U_k$; the $U_k + V_k = X_1 + \ldots + X_k$, job $k + 1$ starts at U_k, an we have:

$$C_{k+1} = \frac{X_1 + \ldots + X_k - D_k}{2} + X_{k+1} \qquad k = 0, \ldots, n - 1, \quad (5)$$

$$D_{k+1} = |X_{k+1} - D_k| \qquad k = 0, \ldots, n - 1. \quad (6)$$

'he flowtime is

$$\sum_{=1} C_k = \frac{n+1}{2} X_1 + \frac{n}{2} X_2$$

$$+ \ldots + \frac{2}{2} X_n - (D_0 + \ldots + D_{n-1})/2 \qquad (7)$$

nd, since $E(X_j) = 1$ for $j = 1, \ldots, n$,

$$\left(\sum_{k=1}^{n} C_k\right) = \frac{n(n+3)}{4} - \frac{1}{2} E\left(\sum_{k=0}^{n-1} D_k\right), \qquad (8)$$

) to minimize expected flowtime we want to maxi- iize the expectation of $\sum_{k=0}^{n-1} D_k$.

It is easy to verify from (6) that if D_0 belongs to ass I′, II′ or III′ and X_j for $j = 1, \ldots, n$ belong to ie corresponding class I, II or III, then D_k for $k = 1,$., n belong to the same class I′, II′ or III′ with iixing probabilities $1 - \alpha_k - \beta_k$, α_k, β_k. For classes , II′, or III′, define χ_k and δ_k as 0, 1, 2 according to hether X_k and D_k are distributed as 0 or Y or Z, spectively, Note that

$$(D_k) = E(E(D_k \mid \delta_k)) = E(\delta_k) = \alpha_k + 2\beta_k. \qquad (9)$$

'e will now treat each class separately.

lass I. Here D_k and X_{k+1} are each a mixture of ;ponential random variables with means 0, 1, 2. ephrase (6) as

$$_{k+1} = \max(D_k, X_{k+1}) - \min(D_k, X_{k+1}). \qquad (10)$$

ecall that if Y, Z are exponential with rates λ_1, λ_2 nd means $1/\lambda_1$, $1/\lambda_2$) then $\min(Y, Z)$ is exponential ith rate $\lambda_1 + \lambda_2$, and $\max(Y, Z) - \min(Y, Z)$ independent of $\min(Y, Z)$ and is distributed as ;p(λ_1) with probability $\lambda_2/(\lambda_1 + \lambda_2)$ and as exp(λ_2) ith probability $\lambda_1/(\lambda_1 + \lambda_2)$. Conditional on the ilue of χ_{k+1} and δ_k we find that β_{k+1}, the value of $(\delta_{k+1} = 2)$, is

$${}_{+1} = \begin{array}{c|ccc} & & \chi_{k+1} & \\ \delta_k & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2/3 \\ 2 & 1 & 2/3 & 1 \end{array} . \qquad (11)$$

lso, for $\delta_k = 0$ to hold it is necessary that $\delta_0 = \chi_1 =$. $= \chi_k = 0$, so, letting $p_0 = 1 - \alpha_0 - \beta_0$,

$$- \alpha_k - \beta_k = P(\delta_k = 0) = p_0 p_1 \ldots p_k. \qquad (12)$$

Combining (11) and (12), we obtain a difference equation

$$\beta_{k+1} = 2/3\, \beta_k + 2/3\, p_{k+1} + 1/3\, p_0 \ldots p_{k+1}, \qquad (13)$$

which is solved by

$$\beta_k = \left(\frac{2}{3}\right)^k \beta_0 + \frac{1}{3} \sum_{j=1}^{k} (2p_j + p_0 \ldots p_j)\left(\frac{2}{3}\right)^{k-j}$$

$$k = 1, \ldots, n. \qquad (14)$$

Substituting (14) in

$$E(D_k) = 1 - p_0 \ldots p_k + \beta_k \qquad (15)$$

and adding, we obtain

$$E\left(\sum_{k=0}^{n-1} D_k\right) = n + 3\left(1 - \left(\frac{2}{3}\right)^n\right)\beta_0 - p_0 + 2\sum_{k=1}^{n-1} p_k$$

$$- \sum_{k=1}^{n-1} (2p_k + p_0 \ldots p_k)\left(\frac{2}{3}\right)^{n-k}. \qquad (16)$$

Consider now a pairwise exchange between p_k and p_{k+1}, $1 \leq k \leq n-1$. Let Δ denote the value of (16) minus the expected value when p_k and p_{k+1} are exchanged. Then

$$\Delta = (2/3)^{n-k}(1 - p_0 \ldots p_{k-1})(p_k - p_{k+1})$$

$$k = 1, \ldots, n-1. \qquad (17)$$

We see that $\Delta < 0$ if $p_{k+1} > p_k$, and $E(\sum_{k=0}^{n-1} D_k)$ can be increased by the pairwise exchange of p_k and p_{k+1}; the maximal value of $E(\sum_{k=0}^{n-1} D_k)$ is therefore obtained by the schedule that satisfies $p_1 \geq \ldots \geq p_n$. This result proves Theorem 1 for Class I.

Class II. Here X_k and D_k are each a mixture of sums of independent exp(1) random variables of which there are 0 or 1 or 2. Recall that if X, Y are distributed as Erlang with parameters k and l, respectively (i.e., as sums of k and l i.i.d. exponential random variables) with rate 1, then $\max(X, Y) - \min(X, Y)$ is a mixture of Erlangs with parameters $1, \ldots, \max(k, l)$. It is easy to verify that, conditional on the values of χ_{k+1} and δ_k we now get for β_{k+1}, the value of $P(\delta_{k+1} = 2)$, conditional on δ_k and χ_{k+1}:

$$\beta_{k+1} = \begin{array}{c|ccc} & & \chi_{k+1} & \\ \delta_k & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1/2 \\ 2 & 1 & 1/2 & 1/2 \end{array} . \qquad (18)$$

We now proceed as for case I; using (12) and (15) which still hold, and setting up a difference equation for β_k, we obtain

$$\beta_k = \left(\frac{1}{2}\right)^k \beta_0 + \sum_{j=1}^{k} (p_j + p_0 \ldots p_j)\left(\frac{1}{2}\right)^{k-j+1}$$

$$k = 1, \ldots, n, \quad (19)$$

$$\mathrm{E}\left(\sum_{k=0}^{n-1} D_k\right) = n + 2\left(1 - \left(\frac{1}{2}\right)^n\right)\beta_0 - p_0 + \sum_{k=1}^{n-1} p_k - \sum_{k=1}^{n-1} (p_k + p_0 \ldots p_k)\left(\frac{1}{2}\right)^{n-k}. \quad (20)$$

Define Δ as before, as the value of (20) minus the value obtained when p_k, p_{k+1} are exchanged; then

$$\Delta = (1/2)^{n-k}(1 - p_0 \ldots p_{k-1})(p_k - p_{k+1})$$

$$k = 1, \ldots, n-1, \quad (21)$$

and the proof of Theorem 1 for class II follows.

Class III. Let $q_i = 1 - 2p_i$ for $i = 1, \ldots, n$ be the probability that X_i is odd, and let $q_0 = \alpha_0$; we then have

$$\mathrm{E}(D_k \mid D_{k-1} \text{ even}) = 1,$$
$$\mathrm{E}(D_k \mid D_{k-1} \text{ odd}) = 1 - q_k, \quad \text{and}$$
$$\mathrm{E}(D_k) = 1 - \alpha_{k-1} q_k \quad k = 1, \ldots, n. \quad (22)$$

To obtain $\alpha_k = \mathrm{P}(D_k = 1) = \mathrm{P}(D_k \text{ odd})$, we set up the following difference equation:

$$\alpha_k = \alpha_{k-1}(1 - q_k) + (1 - \alpha_{k-1})q_k. \quad (23)$$

We see immediately that

$$\alpha_0 = q_0,$$
$$\alpha_1 = q_0 + q_1 - 2q_0q_1, \text{ and}$$
$$\alpha_2 = q_0 + q_1 + q_2 - 2q_0q_1 - 2q_0q_2 - 2q_1q_2 + 4q_0q_1q_2.$$

Letting $S_r^{(k+1)}$ denote the rth symmetric function of $q_0, \ldots, q_k$, that is,

$$S_r^{(k+1)} = \sum_{0 \leq i_1 < \ldots < i_r \leq k} \cdots \sum q_{i_1} \ldots q_{i_r} \quad r = 1, \ldots, k+1,$$

we find, by induction, the solution to (23) as

$$\alpha_k = \sum_{r=1}^{k+1} (-2)^{r-1} S_r^{(k+1)}. \quad (24)$$

Substituting in (22) and adding up over $k = 0, \ldots, n-1$, we get

$$\mathrm{E}\left(\sum_{k=1}^{n-1} D_k\right) = (n-1) - \sum_{r=2}^{n} (-2)^{r-2} S_r^{(n)}. \quad (25)$$

We see that $\mathrm{E}(\sum_{k=1}^{n-1} D_k)$ is a symmetric function ($q_0, q_1, \ldots, q_{n-1}$ and any pairwise exchanges, or an permutations of $q_1, \ldots, q_{n-1}$, will not affect (25 Thus the order of the jobs $1, \ldots, n-1$ has no effe on the flowtime. The only decision to be made which job to do last. We take a derivative with respe to q_{n-1} to get

$$\frac{d}{dq_{n-1}} \mathrm{E}\left(\sum_{k=1}^{n-1} D_k\right) = -\sum_{r=1}^{n-1} (-2)^{r-1} S_r^{(n-1)} = -\alpha_{n-2} < 0. \quad (2\!$$

Therefore, if $p_n > p_{n-1}$, so that $1 - 2p_n = q_n < q_{n-1}$, pairwise exchange between p_{n-1} and p_n will increas $\mathrm{E}(\sum_{k=1}^{n-1} D_k)$. By the symmetry, we can make a simila improvement if $p_n > p_k$ for any $k = 1, \ldots, n-$ Hence the maximum is achieved when $p_n =$ $\min_{1 \leq k \leq n} p_k$. We have shown that the expected flow time depends only on the last job in the schedule, an is maximized if that job has smallest variance. Thu there are many optimal schedules. One of these opt mal schedules is LVF. This result completes the proc of Theorem 1.

Corollary 1. *For $m = 2$ machines, LVF minimiz the expected makespan for classes I, II and III.*

Proof. The makespan $C_{\max}$ is given by

$$C_{\max} = \frac{X_1 + \ldots + X_n + D_n}{2}, \quad (2$$

so we need to minimize $\mathrm{E}(D_n)$. For class I, by (14) an (15),

$$\mathrm{E}(D_n) = 1 + \left(\frac{2}{3}\right)^n \beta_0 - p_0 \ldots p_n + \frac{1}{3}\sum_{k=1}^{n} (2p_k + p_0 \ldots p_k)\left(\frac{2}{3}\right)^{n-k}. \quad (2\!$$

For class II, by (15) and (19),

$$\mathrm{E}(D_n) = 1 + \left(\frac{1}{2}\right)^n \beta_0 - p_0 \ldots p_n + \frac{1}{2}\sum_{k=1}^{n} (p_k + p_0 \ldots p_k)\left(\frac{1}{2}\right)^{n-k}. \quad (2\!$$

It is easy to check by a pairwise exchange of p_k an p_{k+1} that the corollary holds.

For class III, we note that, as in (25), we have

$$\mathrm{E}\left(\sum_{k=1}^{n} D_k\right) = n - \sum_{r=2}^{n+1} (-2)^{r-2} S_r^{(n+1)}. \quad (30$$

This expression is symmetric in $q_1, \ldots, q_n$, and there fore independent of the schedule. On the other hand

LVF (or any rule that puts $p_n = \min_{1 \leqslant k \leqslant n} p_k$), maximizes $E(\sum_{k=1}^{n-1} D_k)$. Hence, LVF minimizes $E(D_n)$, and the corollary is proved.

3. Minimization of Expected Makespan

In this section we prove Theorem 2 for classes I and II for any number of machines m. The proof for class III with $m = 2$ was given in Corollary 1. Assume the m machines are available originally at times $U_0^{(1)} \leqslant \ldots \leqslant U_0^{(m)}$, with $\sum U_0^{(i)} = 0$, and with $U_0^{(i)}$ in class I′ or II′ for $i = 1, \ldots, m$, according to whether the X_j's are in class I or II. Consider jobs $1, \ldots, k$ and let $U_k^{(1)} \leqslant \ldots \leqslant U_k^{(m)}$ be the times at which the machines complete their processing, so that

$$\sum_{i=1}^{m} U_k^{(i)} = X_1 + \ldots + X_k. \tag{31}$$

Define

$$D_k^{(i)} = U_k^{(i+1)} - U_k^{(i)} \quad i = 1, \ldots, m-1, \qquad B_k^{(i)} = U_k^{(m)} - U_k^{(m-i)}. \tag{32}$$

Clearly $C_{\max} = U_n^{(m)}$, and therefore the makespan is

$$C_{\max} = \frac{X_1 + \ldots + X_n + B_n^{(1)} + \ldots B_n^{(m-1)}}{m} = X_1 + \ldots + X_n + D_n^{(1)} + 2D_n^{(2)} + \ldots + (m-1)D_n^{(m-1)}. \tag{33}$$

We will show that exchanging p_k and p_{k+1} when $p_k < p_{k+1}$ will lead to all of $D_j^{(1)}, \ldots, D_j^{(m-1)}$ decreasing stochastically for $j = k+1, \ldots, n$, from which the theorem will follow.

Consider again the schedule of jobs $1, \ldots, k$. At time $U_k^{(1)}$ one machine completes a job and is no longer occupied by any of jobs $1, \ldots, k$. The other $m - 1$ machines are still occupied by the (> 0) remainders of some of the jobs $1, \ldots, k$, whose distributions belong to class I′ or II′. Let $\delta_k^{(1)}, \delta_k^{(2)}, \ldots, \delta_k^{(m-1)}$ be defined as before for the $m - 1$ remainder jobs, and define N_k as the number of $\delta_k^{(j)}$'s which are equal to 2, for $k = 0, 1, \ldots, n$.

Proposition. *For jobs of class I or II, let*

$N' = N_{k+1} \mid N_{k-1} = N,\ \chi_k = 2,\ \chi_{k+1} = 1,$ *and*

$N'' = N_{k+1} \mid N_{k-1} = N,\ \chi_k = 1,\ \chi_{k+1} = 2;$

then $N'' \geqslant_{st} N'$.

Proof. Under these conditioning events, N_{k+1} can assume values $N - 1$, N or $N + 1$. For class I, straightforward calculation (Figure 1) shows

$$\begin{aligned} p(N' = N+1) &= \left(\frac{2m-2N-2}{2m-N-1}\right)^2 \\ &\leqslant \frac{2m-2N}{2m-N} \cdot \frac{2m-2N-2}{2m-N-1} \\ &= P(N'' = N+1) \quad \text{and} \\ P(N' = N-1) &= \frac{(N+1)}{2m-N-1}\left(\frac{N}{2m-N}\right) \\ &\geqslant \left(\frac{N}{2m-N}\right)^2 = P(N'' = N-1), \end{aligned} \tag{34}$$

and the proposition follows.

The proof for class II is less direct. Recall that, for class II, if χ_k is 1, X_k is exp(1) while if $\chi_k = 2$, X_k is erlang(2), i.e., a sum of two independent exp(1) random variables. So altogether, for N' and N'', X_k and X_{k+1} consist of 3 independent exp(1) random variables, each of the jobs k and $k + 1$ starts with an exp(1) duration (let X, Y denote these durations), and the third exp(1) random variable (let Z denote its duration) is added, after the initial exp(1) duration to (i) job k in the case of N', and (ii) job $k + 1$ in the case of N''. We now use a sample path argument to show that $N' \leqslant_{ST} N''$. At $U_{k-1}^{(1)}$ we start X together with whatever $m - 1$ remainders are on the other machines. If X is not the first to end, so that another remainder job ends first, we start Y, together with X and $m - 2$ other job remainders; in this case the rest of the schedule has the same stochastic behavior for both N' and N'' and they have the same distribution. If X ends first, we start Z in the realization of N' and Y in the realization of N''. We continue the sample path argument with the assumption that Z in N' equals Y in N''. Following X we now have Z (or Y) running together with $m - 1$ other job remainders. If Z (and Y) end first, the rest of the schedule has the same stochastic behavior for both N' and N'' and they have the same distribution. If, however, another job ends first, then in the realization of N' we start Y immediately in parallel with Z and $m - 2$ other job remainders, while in the realization of N'', Y is running in parallel to the other $m - 2$ job remainders, and Z is waiting for the completion of Y before it can start.

Let $\tilde{N}$ be the number of job remainders out of the $m - 2$ for which $\chi = 2$; then we have $N'' = \tilde{N} + 1$ while $N' \leqslant \tilde{N}$, and the proposition is proved.

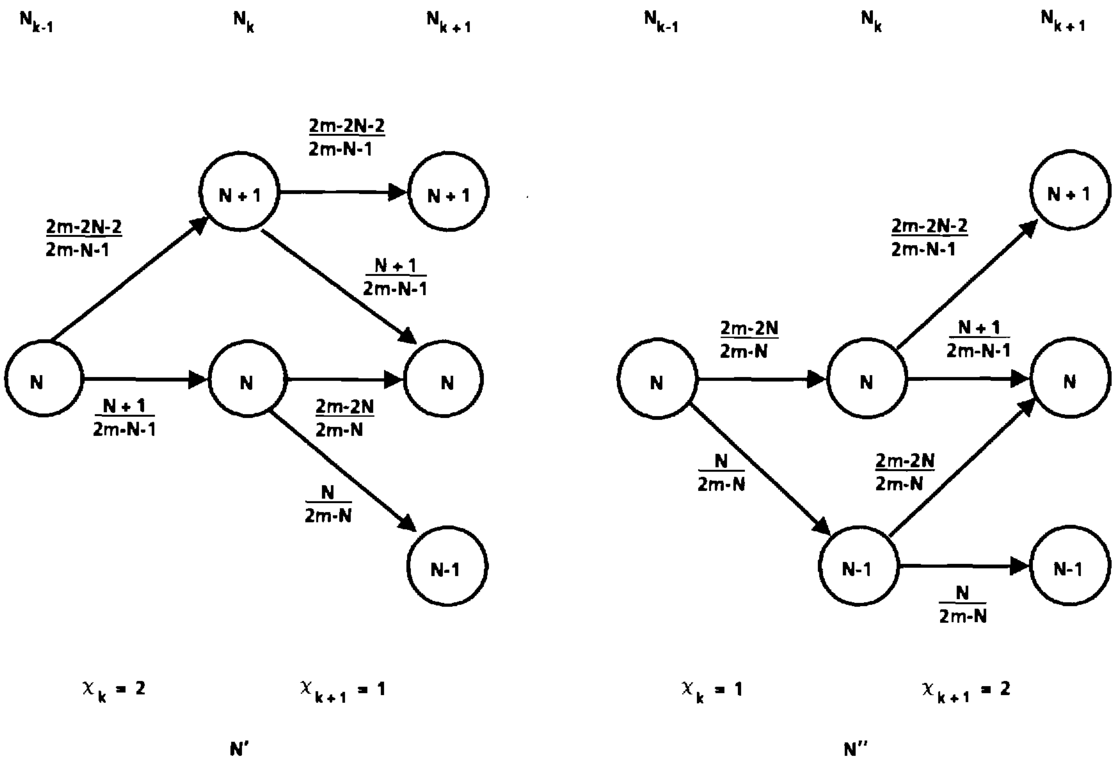


Figure 1. The distribution of N', N''.

This proposition supplies the crucial exchange argument to prove the theorem. Consider changing the order of jobs k and $k + 1$. Doing so will affect the makespan only if one of the jobs has $\chi = 2$ and the other $\chi = 1$. Doing k followed by $k + 1$ we have $\chi_k = 2$, $\chi_{k+1} = 1$, and N_{k+1} distributed like N' with probability $p_k(1 - 2p_{k+1})$; the probability that N_{k+1} is distributed like N'' is analogously $p_{k+1}(1 - 2p_k)$. Reversing the order will add $p_{k+1} - p_k$ to the probability of the first event and decrease the probability of the second event by the same amount. Thus, if $p_{k+1} > p_k$, reversing the order will decrease N_{k+1} stochastically. It is easy to see that $D_{k+1}^{(1)}$ is stochastically increasing in N_{k+1}. So also is N_{k+2}, and so are the numbers of job remainders with $\chi = 2$ when $1, 2, \ldots$ of the $m - 1$ jobs remaining at $U_{k+1}^{(1)}$ are completed. This result proves that the decrease in N_{k+1} will cause $D_j^{(i)}$ for $k + 1 \leq j \leq n$ and $1 \leq i \leq m - 1$ to decrease stochastically. We thus complete the proof.

4. Generalizations

In this section we comment on the results of the previous sections and point out some possible extensions.

(a) Preemptive scheduling of class I and II. To minimize expected flowtime, it is clearly optimal to locate and dispose of zero duration jobs first, at no cost. If we were to minimize makespan, zero duration jobs would be irrelevant and could be scheduled anywhere; we can therefore assume we know which are zero duration jobs, and ignore them. The remaining jobs will then have $\chi = 1$ or $\chi = 2$, and will correspond to the model of Weber. The ensuing policies are highest-hazard-rate-first for flowtime, lowest-hazard-rate-first for makespan. These policies differ from the LVF rule.

(b) Dependence of expected flowtime and makespan on $p_1, \ldots p_n$.

Corollary 2. *For every permutation schedule, the expected flowtime is decreasing in each p_j for $j = 1, \ldots,$ n for $m = 2$ machines and jobs of classes, I, II and III. For every permutation schedule the expected makespan is increasing in each p_j for $j = 1, \ldots, n$ for m machines, and for jobs in classes I and II. The expected makespan is also increasing in p_n, where n is the last job, for jobs in class III and for $m = 2$ machines.*

Proof. The proof is conducted by taking derivatives of (16, 20, 26, 28, 29); we omit the details. The dependence of expected makespan on $p_1, \ldots, p_{n-1}$ for jobs in class III with $m = 2$ can also be fully analyzed, but it is not simply monotone.

(c) Generalization of class I and class II. Theorems 2 and 3 can be proved for a family of jobs $X_1, \ldots, X_n$ whenever all the X_j's are a mixture of 0, Y, Z and have the same mean $E(X_j) = \theta$, and either $Y \sim \exp(1)$, $Z \sim \exp(\lambda)$, $0 < \lambda < 1$, or $Y \sim \exp(1)$, $Z \sim \exp(1) * \exp(\lambda)$, $\lambda > 0$.

5. LVF Is Not Optimal for More General 3-Valued Distributions

In this section we examine the case in which the X_j belong to a class of distributions that generalize class III. We find the policy that minimizes both expected flowtime and expected makespan; surprisingly, this policy is not LVF. Similar optimal policies for some other models have been obtained by Pinedo (1981) and Weiss (1984).

We take fixed θ, $-1 < \theta < 1$ and p_j, $q_j = 1 - 2p_j$, $0 < q_j < 1 - |\theta|$ for $j = 1, \ldots, n$ and let

$$X_j = \begin{cases} 0 & \text{w.p.} \quad p_j + \dfrac{\theta}{2} = \dfrac{1 - q_j + \theta}{2}. \\ 1 & \text{w.p.} \quad 1 - 2p_j = q_j. \\ 2 & \text{w.p.} \quad p_j - \dfrac{\theta}{2} = \dfrac{1 - q_j - \theta}{2}. \end{cases} \tag{35}$$

For this class we have

$$E(X_j) = 1 - \theta, \text{ and}$$
$$V(X_j) = 1 - q_j - \theta^2. \tag{36}$$

We note that if $p_k \geq p_j$, then $X_k \geq_v X_j$, that is, X_k is stochastically more variable than X_j (see Ross 1983, Section 8.5 for definition).

We will establish the following theorem.

Theorem 3. *For $m = 2$ machines, and for $X_1, \ldots, X_n$ distributed as (35), all with mean $1 - \theta$, and with $p_1 \geq p_2 \geq \ldots p_n$, the following rule minimizes expected makespan and the expected flowtime:*

a) *If $\theta > 0$, so that $E(X_j) < 1$, LVF is optimal—that is, the schedule $1, \ldots, n$ is optimal.*
b) *If $\theta = 0$, any policy that has n last is optimal.*
c) *If $\theta < 0$, so that $E(X_j) > 1$, the optimal schedule has the following structure: smallest variance last, largest variance before last, second smallest variance second before last, second largest variance third before last, and so forth—that is, the schedule $\ldots, 2, n-1, 1, n$ is optimal.*

Proof. Let $D_0, D_1, \ldots, D_n$ be defined as in the proof of Theorem 1, with D_k in class III′:

$$D_k = \begin{cases} 0 & \text{w.p.} \quad \gamma_k + \dfrac{\eta_k}{2} = \dfrac{1 - \alpha_k + \eta_k}{2}, \\ 1 & \text{w.p.} \quad 1 - 2\gamma_k = \alpha_k, \\ 2 & \text{w.p.} \quad \gamma_k - \dfrac{\eta_k}{2} = \dfrac{1 - \alpha_k - \eta_k}{2}, \end{cases} \tag{37}$$

with initial values for D_0 given by $q_0 = \alpha_0$ and η_0, so that

$$E(D_k) = 1 - \eta_k, \quad \text{and}$$
$$V(D_k) = 1 - \alpha_k - \eta_k^2. \tag{38}$$

The following difference equations can be set up for α_k and η_k:

$$\alpha_k = \alpha_{k-1} + q_k - 2\alpha_{k-1}q_k, \quad \text{and} \tag{39}$$

$$\eta_k = \alpha_{k-1}q_k + \theta\eta_{k-1}. \tag{40}$$

Equation 39 is the same as (23), and has (24) as its explicit solution, in terms of $q_0, \ldots, q_n$; Equation 40 is solved by

$$\eta_k = \theta^k\eta_0 + \theta^{k-1}\alpha_0 q_1 + \theta^{k-2}\alpha_1 q_2 + \ldots + \alpha_{k-1}q_k, \tag{41}$$

and we get

$$E(D_n) = 1 - \theta^n\eta_0 - \sum_{j=1}^{n} \theta^{n-j}\alpha_{j-1}q_j, \tag{42}$$

$$E\left(\sum_{j=0}^{n-1} D_j\right) = n - \frac{1 - \theta^n}{1 - \theta}\eta_0 - \sum_{j=1}^{n-1} (1 - \theta^{n-j})\alpha_{j-1}q_j. \tag{43}$$

Recall that to minimize expected makespan we need to minimize $E(D_n)$, and to minimize expected flowtime we need to maximize $E(\sum_{j=0}^{n-1} D_j)$.

Let $\Delta_F(k, l)$ be the value of $E(\sum_{j=0}^{n-1} D_j)$ minus the value with the order of jobs k and l exchanged, and let $\Delta_M(k, l)$ be similarly defined for $E(D_n)$. A schedule for which $\Delta_F(k, l) < 0$ ($\Delta_M(k, l) > 0$) for some pair k and l cannot be optimal for flowtime (makespan). Hence, $\Delta_F(k, l) \geq 0$ ($\Delta_M(k, l) \leq 0$) for all k and m is a necessary condition for minimal expected flowtime (makespan). Using (42, 43) we obtain the following results:

$$\Delta_F(k, k+1) = \theta^{n-k-1}(q_{k+1} - q_k)\alpha_{k-1} \qquad k = 1, \ldots, n-1, \tag{44}$$

$$\Delta_F(k-1, k+1) = \theta^{n-k-1}(q_{k+1} - q_{k-1})(\alpha_{k-2}(1 - q_k + \theta) + q_k(1 - \alpha_{k-2})) \qquad k = 2, \ldots, n-1, \tag{45}$$

and

$$\Delta_M(k, l) = -(1-\theta)\Delta_F(k, l). \quad (46)$$

For $\theta > 0$, (44) implies that $q_n \geq q_{n-1} \geq \ldots \geq q_1$, is necessary for flowtime optimality, and, since this condition uniquely determines the schedule, and an optimum exists, it is sufficient; this result proves (a) for flowtime.

For $\theta < 0$, (44) and (45) imply the following optimal orders:

For n even, $q_n \geq q_{n-2} \geq \ldots \geq q_2 \geq q_1 \geq q_3 \geq \ldots q_{n-3} \geq q_{n-1}$.

For n odd, $q_n \geq q_{n-2} \ldots \geq q_3 \geq q_1 \geq q_2 \geq \ldots q_{n-3} \geq q_{n-1}$.

This conclusion proves (c) for flowtime.

The proof of (a) and (c) for makespan follows from (46). (b) was proved in Section 2.

Acknowledgment

This work was supported in part by National Science Foundation grants ECS-83-1223 (Professor Pinedo) and ECS-87-12798 (Professor Weiss).

References

BRUNO, J., P. DOWNEY AND G. FREDERICKSON. 1981. Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan. *J. Assoc. Comput. Mach.* **28,** 100–113.

GLAZEBROOK, K. D. 1979. Scheduling Tasks with Exponential Service Times on Parallel Processors. *J. Appl. Prob.* **16,** 685–689.

PINEDO, M. 1981. Minimizing Makespan with Bimodal Processing Time Distributions. *Mgmt. Sci.* **27,** 582–587.

PINEDO, M., AND G. WEISS. 1979. Scheduling Stochastic Tasks on Two Parallel Processors, *Naval Res. Logist. Quart.* **26,** 527–536.

PINEDO, M., AND S. H. WIE. 1984. Inequalities for Stochastic Flow Shops and Stochastic Job Shops. (submitted to *Naval Research Logistics Quarterly*).

ROSS, S. M. 1983. *Stochastic Processes.* John Wiley & Sons, New York.

WEBER, R. R. 1982. Scheduling Jobs with Stochastic Processing Requirements on Parallel Machines to Minimize Makespan or Flow Time. *J. Appl. Prob.* **19,** 167–182.

WEBER, R. R., P. VARAIYA AND J. WALRAND. 1986. Scheduling Jobs with Stochastically Ordered Processing Times on Parallel Machines to Minimize Expected Flowtime. *J. Appl. Prob.* **23,** 841–847.

WEISS, G. 1984. Scheduling Spares with Exponential Lifetimes in a Two-Component Parallel System. *Naval Res. Logist. Quart.* **31,** 431–446.

WEISS, G., AND M. PINEDO. 1980. Scheduling Tasks with Exponential Service Times on Nonidentical Processors to Minimize Various Cost Functions. *J. Appl. Prob.* **17,** 187–202.

Probability in the Engineering and Informational Sciences, 2, 1988, 269–278. Printed in the U.S.A.

BRANCHING BANDIT PROCESSES

GIDEON WEISS
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205
and
Department of Statistics
Tel Aviv University

A set of n_i arms of type i, $i = 1, \ldots, L$, is available. A pull of arm of type i occupies a duration V_i at the end of which a reward C_i and $N_{i1}, \ldots, N_{iL}$ new arms are obtained, while all other arms are frozen. A Gittins priority order of types is obtained and shown to yield the maximal discounted reward from this branching process of arms.

1. INTRODUCTION

In the classic multiarmed bandit problem there is a fixed set of N arms, which are in states $X_1(t), \ldots, X_N(t)$. At time t one of the arms, say $i = i(t)$, is chosen and pulled. This yields an immediate reward, $R(t) = R_i(X_i(t))$, and a Markovian transition of arm i to state $X_i(t+1)$, depending on $X_i(t)$ alone. Meanwhile, all other arms are frozen at $X_j(t+1) = X_j(t)$, $j \neq i(t)$. The objective is to choose the arms $i(1), \ldots, i(t), \ldots$ so as to maximize $\left\{ \sum_{t=1}^{\infty} \alpha^t R(t) \right\}$. Gittins [2] has shown how to calculate an index $\nu_i = \nu_i(X_i(t))$, which depends on arm i and its state alone, so that the optimal policy is always to choose among $i = 1, \ldots, N$ the arm with the highest index. Many special cases, several extensions, and revisions of the original proof can be found in a series of papers by Gittins and some of his coworkers (Gittins and Glazebrook [4], Gittins and Nash

This research has been partially supported by the National Science Foundation Grant ECS-8712798.

[5], Glazebrook [6], and Kelly [11]). Whittle [19] gives a different proof, and extends [20] the results to an "open" multiarmed bandit process, where in any interval $(t, t+1)$ new arms can arrive, in an arrival stream which is independent and identically distributed (i.i.d.), independent of the control $i(s)$ $s \leq t$. Varaiya et al. [18] give very direct proofs to the original theorems of Gittins and to the "open" process. More recently, Chen and Katehakis [1], Kallenberg [9], and Katehakis and Veinott [10] discuss the calculation of the index, whereas Glazebrook [7] presents a sensitivity analysis. In the present paper, we consider a more general "open" process in which the arrivals depend on the arm which is pulled. We follow throughout the approach used in Meilijson and Weiss [14], which contains results on the average cost criterion. The approach is also similar to that of Varaiya et al. [18].

Consider the following model: Arms can be in one of a finite set of possible states, which we call types; let the type of an arm be denoted by I, $I = 1,2,\ldots,L$. Instead of listing the different arms and their states, the state of the system at any moment is given by $(n_1,\ldots,n_L)$, where n_i is the number of arms of type i present. With an arm of type I, we associate a time V_I, a reward C_I, and arrivals N_{Ij}, $j = 1,\ldots,L$, so that pulling the arm keeps the system busy for a duration V_I, at the end of which a reward C_I is received, and the arm is replaced by an integer number of new arms N_{Ij}, $N_{Ij} \geq 0$, of types j, $j = 1,\ldots,L$. We assume that given I, the durations, the rewards, and the descendants V_I, C_I, N_{Ij}, $j = 1,\ldots,L$, are random variables with arbitrary joint distributions, independent of all other arms, and identically distributed for the same I, $I = 1,\ldots,L$. Starting from some state $(n_1,\ldots,n_L)$, we want to select a type of arm to pull and a policy which will tell us how to continue pulling arms, so as to maximize the sum of discounted expected rewards C_I's obtained over $(0,\infty)$. We shall assume a constant discount rate β, so that a reward C_I received at t is worth $e^{-\beta t}C_I$ now.

In Section 2, we derive some expressions for durations and rewards obtained by following priority rules. In Section 3, we use those to define an index, and prove that the optimal policy is to always pull arms of types with highest index available.

The model of Whittle [20] is a special case of our model in which all the durations V_I are 1, and in which the arrivals N_{Ij} consist of a transition of the arm of type I to an arm of a new type, and of additional arrivals of arms independent of I or of the transition.

2. DISCOUNTED TIMES AND REWARDS OF A PRIORITY POLICY

With an arm of type $i (i = 1,\ldots,L)$ is associated a reward C_i, a nonnegative duration V_i, and a vector of integers N_{ij} of the number of descendants of the arm, of types $j = 1,\ldots,L$. The joint distribution of V_i, N_{ij}, $j = 1,\ldots,L$, is characterized by

$$g_i(s, Z_1,\ldots,Z_L) = E(e^{-sV_i} Z_1^{N_{i1}} \cdots Z_L^{N_{iL}}), \qquad i = 1,\ldots,L. \tag{1}$$

Throughout this section, we shall assume that $1, \ldots, L$ is a priority order, so that if the state at a decision moment is $(n_1, \ldots, n_L)$, the arm chosen is of type

$$i = \min\{j \mid n_j > 0\}.$$

A typical sequence of decisions will start by pulling an arm of type i; for the sake of definiteness, we will assume arms are ordered within types and choose the latest arm of type i. The next decision will occur a time V_i later, from a new state $(N_{i1}, \ldots, N_{ii-1}, n_i - 1 + N_{ii}, n_{i+1} + N_{ii+1}, \ldots, n_L + N_{iL})$. We assume the $N_{i1}, \ldots, N_{iL}$ new arms are again ordered and are all later than the original $n_1, \ldots, n_L$ arms, and when choosing an arm of any type, the latest is chosen (a LIFO policy, within types). If any arms of type 1 were created, the latest is pulled. Arms of type 1 (if there are any) will then be pulled in succession, choosing always the latest one, and creating new arms, for as long as any arms of type 1 are present, or indefinitely. When no more arms of type 1 are present, the latest arm of type 2 is pulled (if there is any), followed again by pulls of arms of type 1; the entire birth process of each arm of type 1 is cleared, moving on to earlier arms until again no more arms of type 1 remain, when the next arm of type 2, the latest created, is pulled. Following the pull of the latest of n_k arms of type k, arms of lower types are pulled until the whole birth process of arms of types $1, \ldots, k-1$ is cleared; then a pull of the latest arm of type k and all its descendants of types $1, \ldots, k-1$ follows, until there are $n_k - 1$ arms of type k left. Only when all arms of type k, and descendants of types $1, \ldots, k$ are cleared, can we move to the latest (if any) arm of type $k+1$, etc.

Consider an ordered pair of types (i,j), where at time $t = 0$ the state consists of a single arm of type i, and consider the sequence of arm pulls starting with the arm of type i, followed by arm pulls of types $1, \ldots, j$ until all arms of types $1, \ldots, j$ are exhausted for the first time, or indefinitely. Call this an (i,j) period. Let T_{ij} be the duration (possibly infinite) of an (i,j) period, and let $M_{ijk}, k = 1, \ldots, L$, be the number of type k arms present at the end of the (i,j) period, where of course $M_{ijk} = 0$, $k \le j$, and where all these arms were created during the (i,j) period. The joint distribution of T_{ij}, M_{ijk}, $k = 1, \ldots, L$, is characterized by

$$G_{ij}(s, Z_1, \ldots, Z_L) = E(e^{-sT_{ij}} Z_1^{M_{ij1}} \cdots Z_L^{M_{ijL}}). \tag{2}$$

Define also

$$G_{io}(s, Z_1, \ldots, Z_L) = g_i(s, Z_1, \ldots, Z_L). \tag{3}$$

Proposition 1: The generating functions G_{ij} satisfy the recursive relation

$$G_{ij}(s, Z_1, \ldots, Z_L) = G_{ij-1}(s, Z_1, \ldots, Z_{j-1}, G_{jj}(s, Z_1, \ldots, Z_L), Z_{j+1}, \ldots, Z_L).$$

(4)

PROOF: The (i,j) period of duration T_{ij} will consist of an $(i,j-1)$ period of duraiton T_{ij-1}, at the end of which there will be M_{ij-1j} type j arms; conditional on $M_{ij-1j}=m$, these give rise to m i.i.d. (j,j) periods. Writing this out we get

$$G_{ij}(s,Z_1,\ldots,Z_L)=E\left[e^{-sT_{ij-1}}\prod_{k>j}Z_k^{M_{ij-1k}}\left\{E\left(e^{-sT_{jj}}\prod_{l>j}Z_l^{M_{jjl}}\right)\right\}^{M_{ij-1j}}\right]$$
$$=G_{ij-1}(s,1,1,\ldots,1,G_{jj}(s,1,\ldots,1,Z_{j+1},\ldots,Z_L),$$
$$Z_{j+1},\ldots,Z_L)$$

which is the required result. ■

The relations (3) and (4) enable us in principle to obtain the generating functions G_{ij} recursively: given G_{ij-1} for all i, we need to solve

$$G_{jj}(s,Z_1,\ldots,Z_L)=G_{jj-1}(s,Z_1,\ldots,Z_{j-1},G_{jj}(s,Z_1,\ldots,Z_L),Z_{j+1},\ldots,Z_L), \tag{5}$$

and then obtain G_{ij} for all i by substituting in Eq. (4).

The random variables T_{ij} and M_{ijk}, $k=1,\ldots,L$, give the duration of the (i,j) period and also the state at the end of the (i,j) period. Let β be the discount rate, let $C_1,\ldots,C_L$ be the rewards of arms of types 1, . . . ,L, and let W_{ij} be the total discounted reward accumulated in the (i,j) period. Define also

$$\gamma_{ij}=E(e^{-\beta T_{ij}})=G_{ij}(\beta,1,\ldots,1),$$
$$\gamma_{io}=E(e^{-\beta V_i}), \tag{6}$$

$$d_{ij}=E(W_{ij}),$$
$$d_{io}=E(W_{io})=E(C_ie^{-\beta V_i}). \tag{7}$$

Proposition 2: The expected discounted reward for period (i,j) satisfies the relations

$$d_{ij}=d_{ij-1}+\frac{d_{jj}}{1-\gamma_{jj}}(\gamma_{ij-1}-\gamma_{ij}), \tag{8}$$

$$d_{jj}=d_{jj-1}\frac{1-\gamma_{jj}}{1-\gamma_{jj-1}}. \tag{9}$$

PROOF: As in the proof of Proposition 1, decompose the (i,j) period into an $(i,j-1)$ period followed by M_{ij-1j} conditionally independent (j,j) periods, which we shall denote by superscripts $1,\ldots,M_{ij-1j}$. Let W_{ij-1} and $W_{jj}^{(1)},\ldots,W_{jj}^{(M_{ij-1j})}$ be the discounted rewards of these periods, discounted to the beginning of these periods. We can write

$$W_{ij}=W_{ij-1}+e^{-\beta T_{ij-1}}\{W_{jj}^{(1)}+e^{-\beta T_{jj}^{(1)}}W_{jj}^{(2)}$$
$$+\cdots+e^{-\beta(T_{jj}^{(1)}+\cdots+T_{jj}^{(M_{ij-1j}-1)})}W_{jj}^{(M_{ij-1j})}\},$$

We note that, conditional on the value of M_{ij-1j}, the pairs $T_{jj}^{(k)}, W_{jj}^{(k)}$ are independent of T_{ij-1}, W_{ij-1}, and are independent of each other for $k = 1, \ldots, M_{ij-1j}$, and are identically distributed with the same joint distribution as the generic pair T_{jj}, W_{jj}. Therefore,

$$\begin{aligned}
d_{ij} &= d_{ij-1} + E[e^{-\beta T_{ij-1}} E\{W_{jj}^{(1)} + e^{-\beta T_{jj}^{(1)}} W_{jj}^{(2)} \\
&\quad + e^{-\beta(T_{jj}^{(1)} + \cdots + T_{jj}^{(M_{ij-1j}-1)})} W_{jj}^{(M_{ij-1j})} | M_{ij-1j}\}], \\
&= d_{ij-1} + E[e^{-\beta T_{ij-1}} d_{jj}(1 + \gamma_{jj} + \cdots + \gamma_{jj}^{M_{ij-1j}-1})], \\
&= d_{ij-1} + \frac{d_{jj}}{1 - \gamma_{jj}} E\{e^{-\beta T_{ij-1}}(1 - \gamma_{jj}^{M_{ij-1j}})\}, \\
&= d_{ij-1} + \frac{d_{jj}}{1 - \gamma_{jj}} \{\gamma_{ij-1} + G_{ij-1}(\beta, 1, \ldots, G_{jj}(\beta, 1, \ldots, 1), 1, \ldots, 1)\}, \\
&= d_{ij-1} + \frac{d_{jj}}{1 - \gamma_{ij}} (\gamma_{ij-1} - \gamma_{ij}).
\end{aligned}$$

Substituting j for i in Eq. (8) and solving, we obtain Eq. (9). ∎

We now define the (i,j) priority index I_{ij} as

$$I_{ij} = \beta \frac{d_{ij}}{1 - \gamma_{ij}}. \tag{10}$$

Note that

$$(1 - \gamma_{ij})/\beta = \{1 - E(e^{-\beta T_{ij}})\}/\beta = E\left(\int_0^{T_{ij}} e^{-\beta t} dt\right).$$

Thus, the index I_{ij} has the typical structure of a Gittins index, in that it is a ratio of expected discounted reward over expected discounted time, calculated for the duration T_{ij}, and relative to the priority order $1, \ldots, L$.

Using Eqs. (8) and (9), we get

$$I_{jj} = I_{jj-1}, \tag{11}$$

$$I_{ij} = \frac{1 - \gamma_{ij-1}}{1 - \gamma_{ij}} I_{ij-1} + \left(1 - \frac{1 - \gamma_{ij-1}}{1 - \gamma_{ij}}\right) I_{jj-1}. \tag{12}$$

In Eqs. (11) and (12), we see that I_{ij} is a convex combination of I_{ij-1} and I_{jj} (or I_{jj-1}). In other words, the reward per unit time in period (i,j) is a weighted average of the reward per unit time in the initial $(i, j-1)$ period and the rewards of the (j,j) periods which follow it. We can reiterate Eq. (12) to get

$$\begin{aligned}
I_{ij} &= \alpha_1 I_{10} + \alpha_2 I_{20} + \cdots + \alpha_j I_{j0} + (1 - \alpha_1 \cdots - \alpha_j) I_{i0}, \\
&= \sum_{k=1}^{j} \alpha_k \frac{\beta E(C_k e^{-\beta V_k})}{1 - E(e^{-\beta V_k})} + (1 - \alpha_1 \cdots - \alpha_j) \frac{\beta E(C_i e^{-\beta V_i})}{1 - E(e^{-\beta V_i})}.
\end{aligned} \tag{13}$$

3. THE GITTINS PRIORITY ORDER AND THE PROOF OF OPTIMALITY

In Section 2, we have taken an arbitrary priority order and defined an index I_{ij} for each pair of types. We now define the optimal priority order, which we call the Gittins order.

DEFINITION: *The Gittins priority order of types is defined by*

$$I_{11} = I_{10} = \max_{1 \le i \le L} I_{i0},$$

$$I_{jj} = I_{jj-1} = \max_{j \le i \le L} I_{ij-1}. \tag{14}$$

Although the definition given by Eq. (14) appears to be implicit, it is obvious that the order can be calculated recursively, where type 1 is obtained by calculating all the I_{i0} and choosing the maximal, and, having defined types $1, \ldots, j-1$, type j is obtained by calculating the values of I_{ij-1} for all i different from $1, \ldots, j-1$, and choosing the maximal; a tie-breaking mechanism for several maxima can be chosen arbitrarily.

We also define the Gittins index of type j as

$$I_j^* = I_{jj}. \tag{15}$$

THEOREM 1: *The priority policy which uses the Gittins priority ordering is optimal among all policies.*

PROOF: Our branching bandit process problem is a semi-Markov decision problem, formulated as follows. The state is the vector of numbers of arms of various types:

$$s = (n_1, \ldots, n_L).$$

The actions available at state s are

$$J(s) = \{ i \mid n_i > 0 \}.$$

The transition from state s under action i will involve a duration

$$T(s,i) = V_i$$

and a new state

$$s' = (n_1 + N_{i1}, \ldots, n_i - 1 + N_{ii}, \ldots, n_L + N_{iL}).$$

A reward C_i will be obtained at the end of $T(s,i)$. All rewards are discounted with continuous discount rate $\beta > 0$.

Consider a general strategy π; let $\pi(s)$ be the process of states and decisions induced by π, for an initial state s, and let $V(\pi)(s)$ be the total expected discounted reward under π, for initial stage s. A nonrandomizing decision function f is a function from the states into the available actions, $f(s) \in J(s)$. A stationary policy is defined by a decision function f, if at all decision moments it chooses $f(s)$; we note that policy by f^∞, and the process it generates for initial state s by $f^\infty(s)$. T is a stopping time for the stationary policy f^∞, if for every initial state s, T is a stopping time of $f^\infty(s)$. We denote by $(f^{(T)}, \pi)$ the policy which does f up to T, and continues with π.

We wish to find an optimal policy π^* such that $V(\pi^*)(s) = \sup_\pi V(\pi)(s)$.

From the theory of Markov decision processes (Ross [16]), because the action space is finite, we know that there exists a stationary optimal policy. Furthermore, f^∞ is optimal if it is excessive, that is, for every decision function g and initial state s,

$$V(g, f^\infty)(s) \le V(f^\infty)(s). \tag{16}$$

To prove excessivity it is enough to proceed as follows: for any g and s_o, if $g(s_o) = f(s_o)$ there is nothing to prove. If $g(s_o) = i \ne f(s_o)$, look at

$$h(s) = \begin{cases} i & i \in J(s) \\ f(s) & \text{otherwise} \end{cases}$$

so that $h(s_o) = g(s_o) = i$. Let T be some (possibly infinite) stopping time, $E(T) > 0$, defined on f^∞. To show Eq. (16) for s_o, it is enough to show that for all s

$$V(h, f^\infty)(s) \le V(f^{(T)}, h, f^\infty)(s), \tag{17}$$

since iterating this m times gives

$$V(h, f^\infty)(s) \le V(f^{(T)m}, m, f^\infty)(s) \underset{n\to\infty}{\rightarrow} V(f^\infty)(s). \tag{18}$$

We now let $1, \dots, L$ be the Gittins priority order, and let f be the priority decision function

$$f(s) = \min(k \mid n_k > 0). \tag{19}$$

We choose as T the stopping time $T_{f(s)f(s)}$ on $f^\infty(s)$; clearly $E(T) \ge \min E(V_k) > 0$.

We now fix i, and show that Eq. (17) holds. If $h(s) = f(s)$ there is nothing to show; otherwise, the state s is

$$\begin{gathered} s = (0, \dots, 0, n_j, \dots, n_i, \dots), \qquad n_i n_j > 0, \\ f(s) = j < i = h(s). \end{gathered} \tag{20}$$

We now compare $V(h, f^\infty)(s)$ to $V(f^{(T)}, h, f^\infty)(s)$. The two processes generated by the policies are described first; see Figure 1.

The policy $(h, f^\infty)(s)$ will act as follows: it will choose the latest arm of type i, then all its descendants of types $\le j$; this will take time T_{ij} with resulting state

$$s^h = (0, \dots, 0, n_j, n_{j+1} + M_{ijj+1}, \dots, n_i - 1 + M_{iji}, \dots).$$

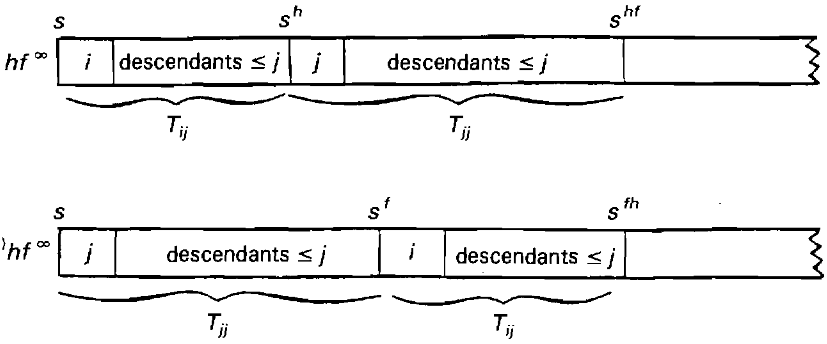


Figure 1. Comparison of two policies.

It will then choose the latest original type j arm, followed by its descendants of types $\leq j$; this will take an additional time T_{jj} with resulting state

$$s^{hf} = (0,\ldots,0,n_j - 1, n_{j+1} + M_{ijj+1} + M_{jjj+1},\ldots,n_i - 1 + M_{iji} + M_{jji},\ldots).$$

From time $T_{ij} + T_{jj}$ and state s^{hf} onwards, the policy will act as f^∞.

The policy $(f^{(T)},h,f)(s)$ will act as follows: it will choose the latest type j arm, and continue f for time T; that is, it will choose all descendants of the j arm of types $\leq j$, with a duration T_{jj} and resulting state

$$s^f = (0,\ldots,0,n_j - 1, n_{j+1} + M_{jjj+1},\ldots,n_i + M_{jji},\ldots).$$

It will then do h, that is, choose a type i arm, and proceed with f^∞; that is, choose all descendants of i of types $\leq j$, with duration T_{ij} and resulting state

$$s^{fh} = (0,\ldots,0,n_j - 1, n_{j+1} + M_{jjj+1} + M_{ijj+1},\ldots,n_i - 1 + M_{jji} - M_{iji},\ldots).$$

From time $T_{jj} + T_{ij}$ and state s^{fh} onwards, it will act as f^∞. Of course, the durations $T_{ij} + T_{jj}$, $T_{jj} + T_{ij}$ and the resulting states s^{hf}, s^{fh} under the two policies have the same distribution. The total discounted rewards of these policies (where we let W denote the random reward of a policy) are:

$$\begin{aligned} W(h,f^\infty)(s) &= W_{ij} + e^{-\beta T_{ij}} W_{jj} + e^{-\beta(T_{ij}+T_{jj})} W(f^\infty)(s^{hf}), \\ W(f^{(T)},h,f^\infty)(s) &= W_{jj} + e^{-\beta T_{jj}} W_{ij} + e^{-\beta(T_{jj}+T_{ij})} W(f^\infty)(s^{fh}). \end{aligned} \tag{21}$$

Taking expectations (with obvious use of independence),

$$\begin{aligned} V(h,f^\infty)(s) &= d_{ij} + \gamma_{ij} d_{jj} + E, \\ V(f^{(T)},h,f^\infty)(s) &= d_{jj} + \gamma_{jj} d_{ij} + E, \end{aligned} \tag{22}$$

where E is the expected value of the last term in Eq. (21), which is the same for both policies. We get (using Eqs. (9), (11), (12), and (14)

$$
\begin{aligned}
V(f^{(T)},h,f^{\infty})(s) - V(h,f^{\infty})(s) &= d_{jj} + \gamma_{jj}d_{ij}, \\
-d_{ij} - \gamma_{ij}d_{jj} &= (1-\gamma_{ij})d_{jj} - (1-\gamma_{jj})d_{ij}, \\
&= (1/\beta)(1-\gamma_{ij})(1-\gamma_{jj})(I_{jj} - I_{ij}), \\
&= (1/\beta)(1-\gamma_{jj})(1-\gamma_{ij-1}) \\
&\quad \times (I_{jj-1} - I_{ij-1}) > 0.
\end{aligned}
$$

Note that in the above proof in Eq. (21), we allow the possibility of $E(T_{jj}) = \infty$, or $P(T_{jj} < \infty) < 1$, or even $P(T_{jj} < \infty) = 0$. This does not affect the proof at all, since $\gamma_{jj} = E(e^{-\beta T_{jj}})$ is still well-defined, with the possibility of $\gamma_{jj} = 0$; as long as $E(T_{jj}) > 0$, $1 - \gamma_{jj} > 0$ so I_{jj} is well-defined; of course, if $\gamma_{jj} = 0$, $I_{jj} = \beta d_{jj}$.

References

1. Chen, Y.R. & Katehakis, M.N. (1986). Linear programming for finite state multiarmed bandit problems. *Mathematics of Operations Research* 11: 180–183.
2. Gittins, J.C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B* 14: 148–177.
3. Gittins, J.C. & Jones, D.M. (1974). A dynamic allocation index for the sequential design of experiments. In J. Gani, K. Sarkadi, & I. Vince (eds.), *Progress in Statistics European Meeting of Statisticians 1972, Vol. 1.* Amsterdam: North-Holland, pp. 241–266.
4. Gittins, J.C. & Glazebrook, K.D. (1977). On Bayesian models in stochastic scheduling. *Journal of Applied Probability* 14: 556–565.
5. Gittins, J.C. & Nash, P. (1977). Scheduling, queues, and dynamic allocation indices. *Proceedings EMS, Prague,* 1974, Czechoslovak Academy of Sciences, Prague, 191–202.
6. Glazebrook, K.D. (1976). Stochastic scheduling with order constraints. *International Journal of Systems Science* 7: 657–666.
7. Glazebrook, K.D. (1987). Sensitivity analysis for stochastic scheduling problems. *Mathematics of Operations Research* 12: 205–223.
8. Harrison, J.M. (1975). Dynamic scheduling of a multiclass queue: discount optimality. *Operations Research* 23: 270–282.
9. Kallenberg, L.C.M. (1986). A note on M.N. Katehakis and Y.R. Chen's computation of the Gittins Index. *Mathematics of Operations Research* 11: 184–186.
10. Katehakis, M.N. & Veinott, A.F. (1987). The multiarmed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12: 262–268.
11. Kelly, F.P. (1981). Multiarmed bandits with discount factor near one: the Bernoulli case. *The Annals of Statistics* 9: 987–1001.
12. Klimov, G.P. (1974). Time sharing service systems I. *Theory of Probability and Applications* 19: 532–551.
13. Klimov, G.P. (1978). Time sharing service systems II. *Theory of Probability and Applications* 23: 314–321.
14. Meilijson, I. & Weiss, G. (1977). Multiple feedback at a single server station. *Stochastic Processes and their Applications* 5: 195–205.
15. Nash, P. & Gittins, J.C. (1977). A Hamiltonian approach to optimal stochastic resource allocation. *Advances in Applied Probability* 9: 55–68.

16. Ross, S.M. (1970). *Applied probability models with optimization applications.* San Francisco, CA: Holden Day.
17. Sevcik, K.C. (1974). Scheduling for minimum total loss using service time distributions. *Journal of the Association for Computing Machine* 21: 66–75.
18. Varaiya, P., Walrand, J., & Buyukkoc, C. (1985). Extensions of the multiarmed bandit problem: the discounted case. *IEEE Transactions on Automatic Control* AC-30: 426–436.
19. Whittle, P. (1980). Multiarmed bandits and the Gittins index. *Journal of the Royal Statistical Society Series B* 42: 143–149.
20. Whittle, P. (1981). Arm acquiring bandits. *Annals of Probability* 9: 284–292.

PEIS 80

Probability in the Engineering and Informational Sciences, 3, 1989, 89–116. Printed in the U.S.A.

SCHEDULING STOCHASTIC JOBS WITH A TWO-POINT DISTRIBUTION ON TWO PARALLEL MACHINES

E.G. COFFMAN, JR.
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

M. HOFRI
Computer Science Department
The Technion
Haifa, Israel

G. WEISS
School of ISyE
Georgia Institute of Technology
Atlanta, Georgia 30320

We analyze the optimal preemptive sequencing of n jobs on two machines to minimize expected total flow time. The running times of the jobs are independent samples from the distribution $Pr(X = 1) = p$, $Pr(X = k + 1) = 1 - p$. We verify that the shortest-expected-remaining-processing-time (SERPT) policy, which is optimal for independent and identically distributed (i.i.d.) running times with a monotone hazard-rate distribution, is not optimal for this distribution. However, we prove that if $p \geq 1/k$, then the number of decisions where SERPT and an optimal policy disagree is bounded by a constant independent of n. For $p < 1/k$, we prove that the *expected* number of such decisions has a similar bound. In addition, bounds on the expected increase in flow times under SERPT are derived; these bounds are also independent of n.

1. INTRODUCTION

We consider an instance of a well-known problem in stochastic scheduling theory: $n \geq 1$ jobs are to be run on one or more identical machines operating in parallel. The running times $X_1, \ldots, X_n$ are not known in advance, but they are known to be independent samples from a given distribution $G(x)$. The problem is to find a preemptive scheduling algorithm that minimizes the expected sum of finishing times (total flow time).

The single-machine problem was solved by Sevick [12] for general $G(x)$. Effectively, a ranking of jobs based on elapsed running time is calculated dynamically. At all times, the job assigned to the machine is one having the least rank. As a result of subsequent extensions to other stochastic optimization problems, this rank is now known as a special case of the Gittins index [6].

Weber [13] solved the problem for two or more machines when $G(x)$ has a monotone hazard rate. He showed that the dynamic SERPT (shortest expected remaining processing time) ordering of job assignments is optimal. This algorithm reduces to either non-preemptive sequencing or processor sharing according to whether the hazard rate is increasing or decreasing, respectively. The multiple-machine problem for general $G(x)$ remains open and appears to be very difficult. In particular, no concrete measure of the complexity of this problem is currently known, e.g., formulating the problem as a "game against nature" [10] or finding an imbedded NP-complete problem have so far been unproductive approaches.

The contribution of this paper is an analysis of the two-machine case for a distribution $G(x)$ with a non-monotone hazard rate; we have chosen a distribution simple enough to be tractable, but one that models interesting practical applications. Specifically, $G(x)$ is the two-point distribution $Pr\{X = 1\} = p$ and $Pr\{X = k + 1\} = 1 - p$ with $0 < p < 1$ and $k \geq 2$ an integer. The hazard rate of $G(x)$ is

$$\frac{Pr\{X = j + 1\}}{Pr\{X > j\}} = \begin{cases} p & j = 0 \\ 0 & 1 \leq j < k \\ 1 & j = k \end{cases}$$

which is neither increasing nor decreasing if $k \geq 2$.

We study scheduling policies satisfying the two properties:

(P1) Neither machine is allowed to remain idle while unfinished jobs remain.

(P2) A job can be preempted only at the point when it has received its first unit of service and requires k units more, i.e., at the earliest point when it becomes known whether a job is *short* (requires only one time unit) or *long* (requires $k + 1$ time units).

It is not difficult to prove that (P1) is not restrictive, i.e., an optimal algorithm with this property can always be found. We suspect that (P2) is also not restrictive. However, a proof of this appears difficult.

In spite of the simplicity of the two-point distribution, subsequent sections show that the optimization problem remains nontrivial. Our main results are strong asymptotic characterizations of optimal and SERPT (shortest expected remaining processing time) policies. For $p \geq 1/k$, we prove a *turnpike* theorem; i.e., whenever the number of remaining, unstarted jobs is sufficiently large, SERPT decisions are optimal. For $p < 1/k$, we show that the expected number of decisions where SERPT is not optimal is bounded by a constant. We also bound the expected difference in flow times produced by optimal and SERPT rules. This asymptotic analysis of an approximation algorithm appears to be new in stochastic scheduling theory and to hold promise for the study of near-optimal algorithms for similar problems.

Our assumptions model inspection/service applications in which customers, devices, etc., are first inspected and then serviced, e.g., repaired, if necessary (an event of probability $1 - p$). The constant inspection time is taken as the unit of time. We approximate the service time k as a fixed multiple of the inspection time, so our model applies primarily to situations in which the service time is large compared to the inspection time. Other potential applications arise in computer systems, where the requirements of users break down into short jobs that can be processed very quickly by the operating system, and long jobs that require the operation of a programing system or a program created by the user. Sevcik [12] mentions other computer scheduling applications.

The literature contains a number of results related to our problem within the general theme of parallel machines, given distributions of independent running times, and the objectives of minimizing either the expected sum (possibly weighted) or the maximum of finishing times. For example, see [1,2,4,7,8, 14,15]. However, virtually all of the results to date are characterized by distributions with monotone (possibly weighted) hazard rates or linear orderings by stochastic dominance. As a consequence, the optimal algorithms have all had a simple ranking structure, as in Sevcik's algorithm, where the rank of a job is determined by a relatively easily computed function of job and machine parameters. The ideas underlying these algorithms for the single-machine case were developed by Gittins [6] and were also suggested by the earlier work of Chazan et al. [3] and Meilijson and Weiss [9] on the scheduling of feedback queues. The results in this paper illustrate the substantial increases in the complexity of scheduling parallel machines, when the above simplifying distributional assumptions do not apply.

The remainder of the paper is organized as follows. In the next section, the mathematical model is formalized. In Section 3, a turnpike theorem for the case $p \geq 1/k$ is proved. Section 4 deals with the case $p < 1/k$, and concluding remarks are given in Section 5.

2. DEFINITIONS

We begin by defining the state space and the corresponding stochastic scheduling process. *Decision states* occur only at integer points when one or both machines finish the first or last time unit of a job. They accumulate all of the available running-time information on which to base a scheduling (assignment) decision. For our problem, decision states are defined by triples (c, I, T), where

(i) $0 \leq c \leq k$ is the number of units of elapsed time already received by the job assigned to the occupied machine, if any. The null value $c = \lambda$ signifies that both machines are available for assignment at a decision point.

(ii) I is a set, possibly empty, of positive integers indexing the unstarted or *new* jobs remaining.

(iii) T is a set, possibly empty, of positive integers indexing the unassigned long jobs having already received their first unit of service (the last k time units of such a job is called a *tail job*). The sets I and T are disjoint.

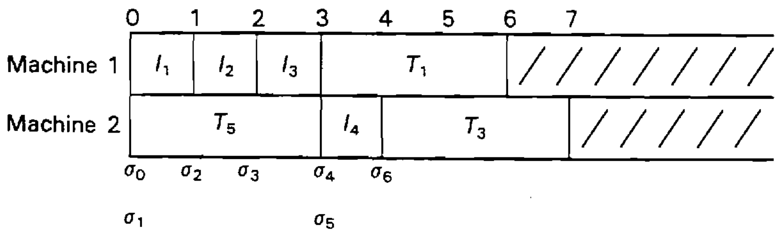
Figure 2.1 illustrates the scheduling process and many of the definitions to follow. Note that I_j denotes the first time unit of X_j, and if X_j is long, then T_j denotes its tail job. The symbol σ is used frequently to denote a decision state.

Observe that if $2 \leq c \leq k$, then the occupied machine is running a tail job with $k - c + 1$ time units left to run; if $c = 1$, then a tail job is just being started on the occupied machine; and if $c = 0$ the first time unit of a new job is just being started on the occupied machine. Thus, decision states in which $c = 0$ or 1 are immediately preceded by decision states in which $c = \lambda$.

The "standard" initial state is $(\lambda, \{1,2, \ldots, n\}, \phi)$ where both machines are available and there are no partially run (i.e., tail) jobs. However, to simplify inductive arguments, it is convenient to allow more general initial states. In Section 3, we allow any state (c, I, T), where $|I| + |T| > 0$ and $c = \lambda$ or $1 \leq c \leq k$; i.e., either both machines are available or one of them is unavailable for the first $k - c + 1$ time units (as if it were finishing some artificial tail job). In Section 4, we also allow $c = 0$, in which case the initial delay on the occupied machine is a random variable: it is 1 with probability p and $k + 1$ with probability $1 - p$. Thus, $c \neq \lambda$ in the initial state may be interpreted as the elapsed time of some artificial new job which is running non-preemptively.

Clearly, the states of the form (c, ϕ, T), $|T| \geq 1$, and (c, I, ϕ), $|I| \geq 1$, allow but one decision: assign a tail or new job, respectively. Final decisions take place in states of the form (c, I, ϕ), $|I| = 1$, or (c, ϕ, T), $|T| = 1$.

In the sample schedules under some policy A, let $F_1, F_2, \ldots, F_{n+r}$ denote the finishing times of the n new jobs and the r tail jobs specified in the initial state $\sigma = (c, I, T)$, $|I| = n$, $|T| = r$. The random variable $A(\sigma) = \sum_{i=1}^{n+r} F_i$ de-



Sample values X_1 X_2 X_3 X_4 (Probability $= p^2(1-p)^2$)

4 1 4 1

T_5 = tail job in initial state

I_i = initial unit of X_i

T_i = tail of $X_i = k = 3$.

Decision states: $\sigma_0 = (\lambda,\{1,2,3,4\},\{5\})$, $\sigma_1 = (1,\{1,2,3,4\},\phi)$,
$\sigma_2 = (2,\{2,3,4\},\{1\})$, $\sigma_3 = (3,\{3,4\},\{1\})$,
$\sigma_4 = (\lambda,\{4\},\{1,3\})$, $\sigma_5 = (1,\{4\},\{3\})$, $\sigma_6 = (2,\phi,($

Flow time = 6 + 2 + 7 + 4 + 3 = 22

Figure 2.1. A sample function for $k = 3$ and initial state $(\lambda,\{1,2,3,4\},\{5\})$.

notes the *total flow time* (or simply *flow time*) under A. OPT refers to a policy that minimizes $EA(\sigma)$, where the expectation is over all $X_1,\ldots,X_n$.

We let f^A denote the decision rule of policy A; i.e., A's decision in state $\sigma = (c,I,T)$ is denoted by $f^A(\sigma)$, which is an integer in I or T. If $i = f^A(\sigma) \in I$, then A makes an *I-decision* and assigns I_i to an available machine; if $i \in T$, then A makes a *T-decision* and assigns T_i to an available machine. (The choice between two available machines can be arbitrary, since it obviously has no effect on flow times.)

Clearly, the order in which new jobs are selected for assignment is unimportant; such jobs are stochastically identical by definition. It is also clear that the expected total flow time starting in state (c,I,T) depends only on c, $|I|$, $|T|$. These observations are exploited in Section 4, where a simpler decision state is adopted. In particular, only the *numbers* of new jobs and tail jobs are combined with the elapsed-time parameter c in a decision state. The more detailed state (c,I,T) is used in Section 3, because it greatly simplifies the proof of the turnpike property for $p \geq 1/k$.

SERPT is an important policy to consider for our problem, because it is optimal when $G(x)$ has a monotone hazard rate, and it is optimal on a single machine when $G(x)$ is our two-point distribution. Specialized to our setup, SERPT reduces either to non-preemptive sequencing (NS) or to preemptive sequencing (PS). According to NS, once a job is assigned, it is run to completion without interruption. In all states (c, I, T), $|I| \geq 1$, policy PS always preempts a long job after its first time unit and assigns a new job in its place.

NS applies under SERPT when $k < EX$ (i.e., when the duration of a tail job is less than the expected total time of a new job), and PS applies when $EX \leq k$. Since $EX = 1 + (1-p)k$, we see that SERPT reduces to NS when $p < 1/k$, and to PS when $p \geq 1/k$.

It is easily verified that SERPT does not minimize expected flow time in general. Indeed, for any $0 < p < 1/2$ and $k > 1$, there are states in which SERPT decisions are not optimal.

Example 1: Consider the initial state $\sigma = (\lambda, \{1\}, \{2,3\})$. If $p < 1/k$, then SERPT reduces to NS and the sample schedules are those in Figure 2.2(a). OPT is shown in Figure 2.2(b) and yields $E[\mathrm{NS}(\sigma) - \mathrm{OPT}(\sigma)] = p(k-1)$. Note that state σ is reached from the standard initial state $(\lambda, \{1,2,3\}, \phi)$ when the first two jobs scheduled are X_2 and X_3, and both are long.

Example 2: Consider the initial state $\sigma = (\lambda, \{1,2\}, \{3\})$ which can be reached from $(\lambda, \{1,2,3,4\}, \phi)$ if I_3 and I_4 are assigned first and one of X_3 and X_4 is long and the other is short. If $p \geq 1/k$, then SERPT reduces to PS, and if in addition $p < 1/2$, then we find $E[\mathrm{PS}(\sigma) - \mathrm{OPT}(\sigma)] = (1-p)(1-2p) > 0$, as shown in Figure 2.3.

There are many other examples showing that SERPT is not optimal. For example, the initial state in Example 1 can be generalized to $\sigma' = (\lambda, \{1\}, T)$ for $|T| \geq 2$ even. For p sufficiently near but still less than $1/k$, we again obtain $E[\mathrm{NS}(\sigma') - \mathrm{OPT}(\sigma')] > 0$. Other examples will be given in the context of the analysis in Section 3.

3. THE OPT AND PS POLICIES FOR $p > 1/k$

Recall that in this section, if (c, I, T) is an initial state, then $c \neq 0$. In the inductive proofs to follow, the size of a state (c, I, T) refers to its position in the lexicographic ordering of $(|I|, |T|)$.

The asymptotic results in this section are considerably more precise than those for NS in Section 4. In particular, we show that, if $p > 1/k$, then all sample schedules under an optimal policy must have the general form illustrated in Figure 3.1. That is, a sample schedule must begin with at least $n - k$ I-decisions in an initial region R_1, end with only T-decisions in a final region R_3, and contain a region R_2 between R_1 and R_3, in which the remaining I-decisions and at most k T-decisions are made. Thus, since PS decisions differ from those of OPT only in the at most k T-decisions in R_2, PS is an optimal turnpike policy.

Example 3: It is tempting, perhaps, to conjecture that the region R_2 always consists of at most one T-decision. The smallest counterexample is provided by the initial state $\sigma = (\lambda, \{3,4,5\}, \{1\})$, which is reached from the standard initial

author: "The" ok here instead of "a"?

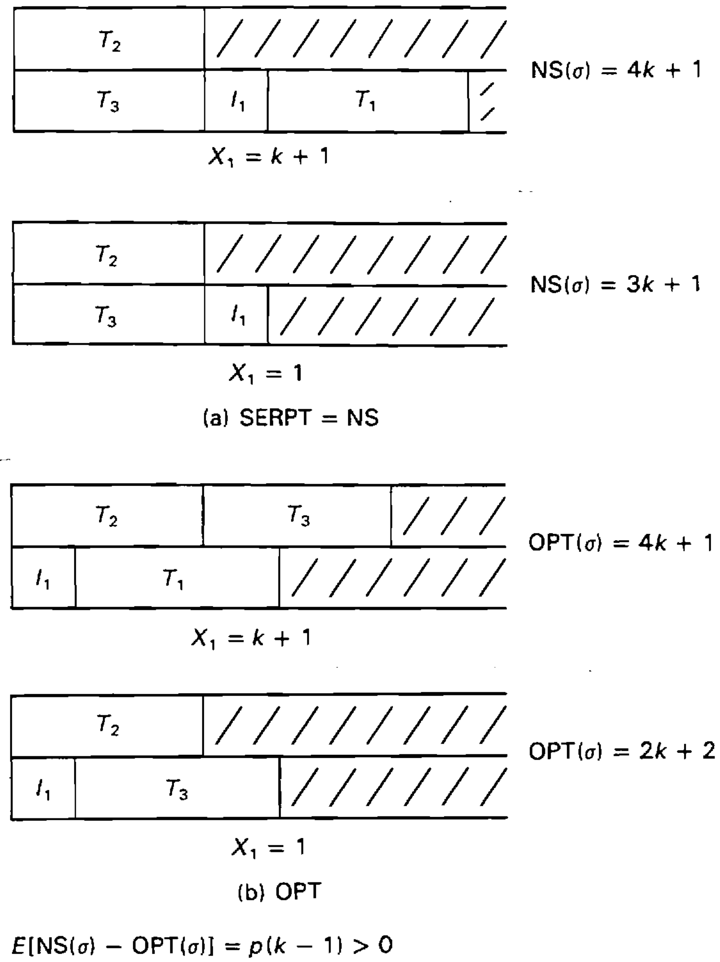
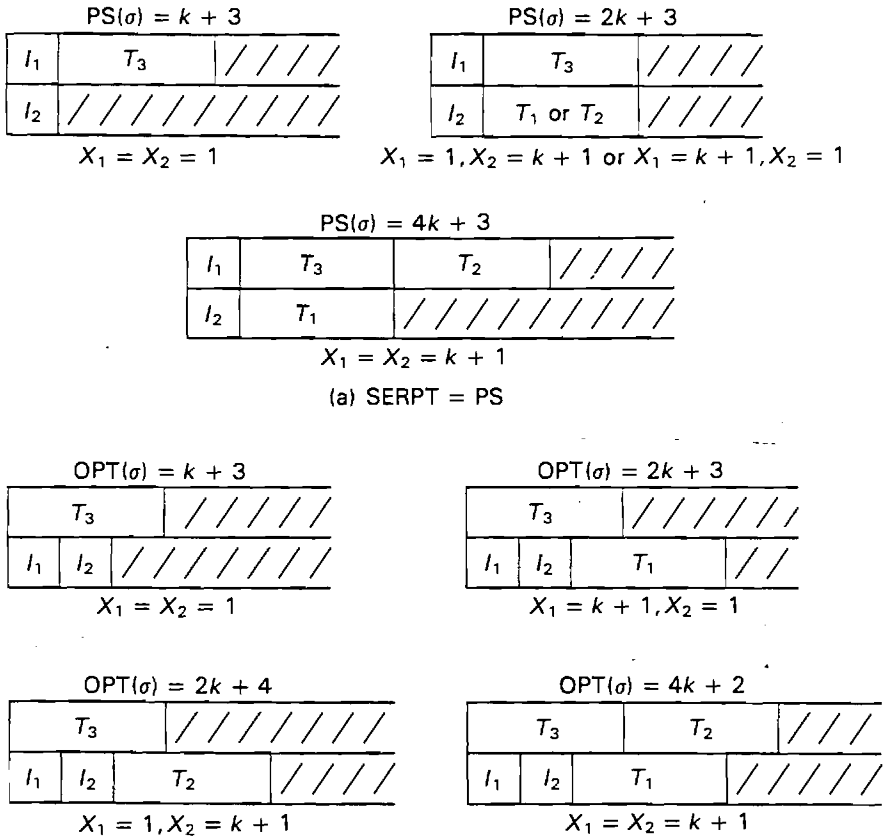


FIGURE 2.2. Counterexample for $p < 1/k$; $\sigma = (\lambda, \{1\}, \{2,3\})$.

state $(\lambda, \{1,2,3,4,5\}, \phi)$ when I_1 and I_2 are run first and X_1 and X_2 are long and short jobs, respectively. An optimal policy can be worked out in the usual way by Bellman equations; a general solution for any initial state can be written, but the expressions are awkward and uninstructive and therefore omitted. A complete analysis shows that for all $1/k < p < 1/(k-1)$, $k \geq 7$, OPT begins with a T-decision on one machine and two consecutive I-decisions on the other. If both of the I-decisions reveal long jobs (thus reaching a state $\sigma' = (3, I', T')$ with $|I'| = 1$ and $|T'| = 2$), OPT again makes a T-decision instead of the one available I-decision. That is, twice, when in a state with an I-decision available (namely, states σ and σ'), OPT chooses a T-decision.



(b) OPT

$E[\mathrm{PS}(\sigma) - \mathrm{OPT}(\sigma)] = (1 - p)(1 - 2p) > 0$

FIGURE 2.3. Counterexample for $1/k \le p < 1/2$; $\sigma = (\lambda, \{1,2\}, \{3\})$.

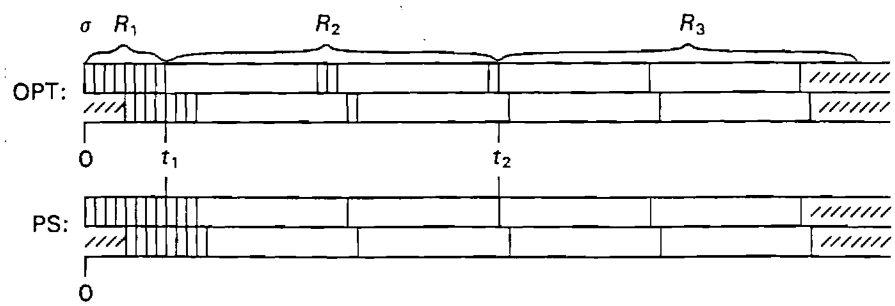


FIGURE 3.1. Sample schedules under OPT and PS.

To prove the properties implicit in Figure 3.1, we begin with two preliminary results having similar, essentially combinatorial proofs. Our first lemma shows that if $p > 1/k$, then OPT never assigns two tail jobs that begin at the same time, unless there is no new job waiting.

LEMMA 3.1: *If $p > 1/k$, then an I-decision is made by OPT in any state $(1, I, T)$ with $|I| > 0$.*

PROOF: The result is obvious if $T = \phi$, so let $|T| \geq 1$ and let $(1, I, T)$ be the smallest state violating the lemma. Without loss of generality, let $\sigma_0 = (1, I, T)$ be the initial state at $t = 0$. At time k, OPT must reach the state $\sigma_1 = (\lambda, I, T - \{j\})$, where $j = f(\sigma_0)$. Since σ_0 is the smallest counterexample, at least one I-decision must be made at time k. Thus, if σ_2 denotes the state following σ_1 at time k, we can assume $i = f(\sigma_2) \in I$, since the ordering of the two assignments at time k can be arbitrary. At the same time, $f(\sigma_1) = l$ with l belonging to $I - \{i\}$ or to $T - \{j\}$.

We now construct a policy A that reverses the order of T_j and I_i, but otherwise preserves the assignments of OPT. Figure 3.2 illustrates the construction.

For A's decision at time 0 in state σ_0, we define the I-decision

$$f^A(1, I, T) = i. \tag{3.1}$$

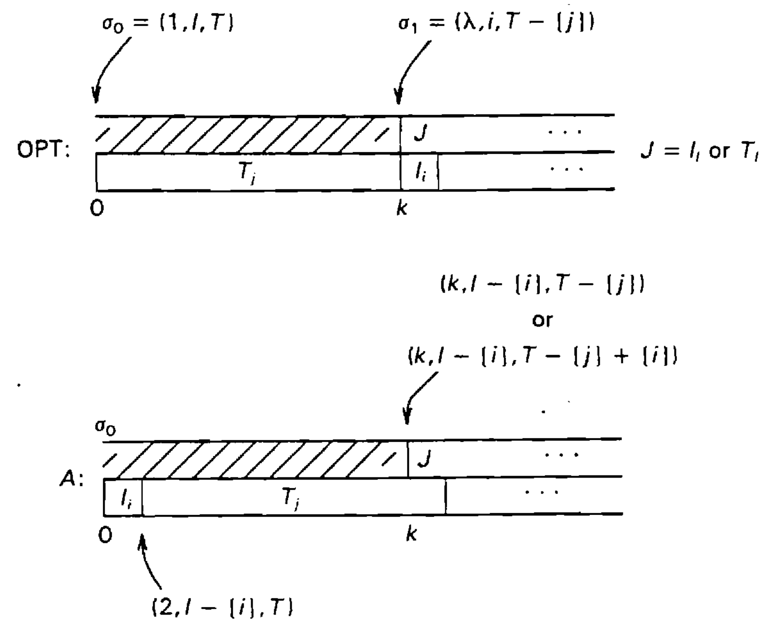


FIGURE 3.2. Comparison for Lemma 3.1.

For A's decision at time 1, we want a T-decision irrespective of whether X_i is short or long; so we define

$$f^A(2, I - \{i\}, T) = f^A(2, I - \{i\}, T + \{i\}) = j. \tag{3.2}$$

Apart from the above interchange of I and T decisions, we want A to simulate OPT. For this purpose, we first set up the appropriate decision at time k, which again must be independent of whether X_i is short or long. We define

$$f^A(k, I - \{i\}, T - \{j\}) = f^A(k, I - \{i\}, T - \{j\} + \{i\}) = l, \tag{3.3}$$

where $l = f(\lambda, I, T - \{j\})$ (either $l \in I$ or $l \in T$). Finally, for states σ reachable from σ_0, other than those in Eqs. (3.1)–(3.3), we define

$$f^A(\sigma) = f(\sigma). \tag{3.4}$$

It is clear from Eqs. (3.1)–(3.3) that in $[0, k+1]$ both A and OPT run I_i, T_j, and one time unit of X_l (either I_l, if $l \in I$, or the first time unit of T_l, if $l \in T$). Thus, the decision state reached by A at the completion of T_j is the same as the decision state reached by OPT at the completion of I_i. By Eqs. (3.1)–(3.4), the only flow times that can differ under A and OPT are those of X_i and X_j. A delays the flow time of X_j by one time unit relative to OPT. But if X_i is short, then OPT delays the flow time of X_i by k time units relative to A. If X_i is long, it has the same flow time under A and OPT. Then

$$E[A(\sigma_0) - \mathrm{OPT}(\sigma_0)] = 1 - pk < 0, \tag{3.5}$$

which contradicts the optimality of OPT. ■

The next result shows that if $p > 1/k$, then OPT cannot make two consecutive T-decisions if the first occurs in a state (c, n, r) with $2 \le c \le k$ and $n \ge 1$. In the remainder of the paper, it is convenient to define

$$c + 1 = \lambda \quad \text{if } c = k, \qquad c - 1 = k \quad \text{if } c = \lambda. \tag{3.6}$$

LEMMA 3.2: *If $p > 1/k$ and OPT makes a T-decision in some state (c, I, T) with $2 \le c \le k$, $|I| \ge 1$ and $|T| \ge 1$, then OPT must make an I-decision in the next state, $k - c + 1$ time units later.*

PROOF: The result is trivial if $|T| = 1$, so let $|T| \ge 2$ and let (c, I, T) be the smallest state violating the lemma. Again, we may suppose without loss of generality that $\sigma_0 = (c, I, T)$ is the initial state. Let $f(\sigma_0) = j$, so that by assumption, in the next state $\sigma_1 = (k - c + 2, I, T - \{j\})$, we have $l = f(\sigma_1) \in T$.

Since σ_0 is the smallest counterexample, OPT must make an I-decision at time k in state $\sigma_2 = (c, I, T - \{j, l\})$. Let $i = f(\sigma_2) \in I$. As in Lemma 3.1, we construct a policy A that reverses the order of a T and I-decision. Figure 3.3 illustrates the construction.

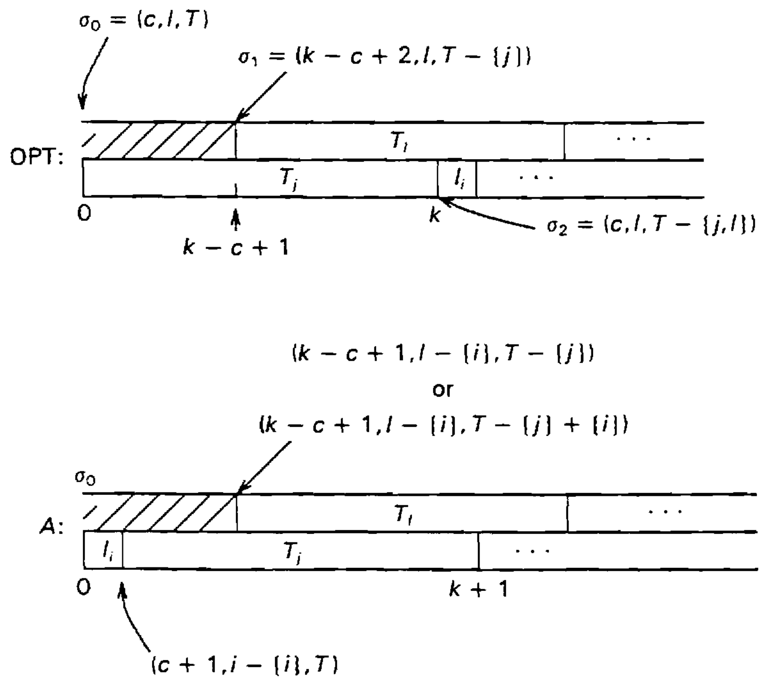


FIGURE 3.3. Comparison for Lemma 3.2.

We define

$$f^A(c,I,T) = i, \tag{3.7}$$

$$f^A(c+1,I-\{i\},T) = f^A(c+1,I-\{i\},T+\{i\}) = j, \tag{3.8}$$

$$f^A(k-c+1,I-\{i\},T-\{j\}) = f^A(k-c+1,I-\{i\},T-\{j\}+\{i\}) = l, \tag{3.9}$$

and for states reachable from σ_0, other than those in Eqs. (3.7)–(3.9), we define

$$f^A(\sigma) = f(\sigma). \tag{3.10}$$

In analogy with Lemma 3.1, Eqs. (3.7) and (3.8) implement the desired interchange of I- and T-decisions, while Eqs. (3.9) and (3.10) assure that A otherwise simulates OPT. It is easy to see that both A and OPT reach the same state at time $k+1$. As in Lemma 3.1, the interchange of I- and T-decisions again produces the contradiction in Eq. (3.5). ■

Our next result shows that if OPT assigns a tail job at some time t on one of the machines, then the time available to run waiting jobs in the interval $[t, t+k]$ on the other machine cannot be less than the number of new jobs remaining at time t; i.e., it must be possible during $[t, t+k]$ to run the initial time units of all new jobs waiting at time t. A weak turnpike property of PS follows immediately from this result: In any state (c, I, T) with $|I| > k$, an I-decision is optimal. After combining this result with Theorem 3.2, we will obtain the stronger property. The number of times OPT makes a non-PS decision (i.e., does not preempt when at least one new job is waiting) is bounded independent of the initial state.

THEOREM 3.1: *If $p \geq 1/k$ and if OPT makes a T-decision in state (c, I, T), then $|I| \leq c-1$ if $c \neq 0$ (by Eq. (3.6), $c-1=k$ if $c=\lambda$), and $|I| \leq k-1$ if $c=0$.*

PROOF: Let (c, I, T) be the smallest state violating the theorem. If $c=0$, then (c, I, T) must be preceded immediately by a state $(\lambda, I+\{l\}, T)$ for some l. In this case, we begin by modifying OPT so that it reverses the decisions in these two states; i.e., $f(\lambda, I+\{l\}, T)=j \in T$ and $f(1, I+\{l\}, T-\{j\})=l$. Clearly, the flow times under OPT are unaffected and the violation of the theorem now occurs in the state $(\lambda, I+\{l\}, T)(|I|>k-1$ becomes $|I+\{l\}|>k)$. It follows that we may assume an algorithm OPT such that $\sigma_0=(c, I, T)$, $c \neq 0$, is the smallest *initial* state violating the theorem; i.e., $|I|>c-1$ and $|T| \geq 1$.

Since we are assuming $|I|>c-1$ and a T-decision at time 0, there must be at least one I-decision at time k in every sample schedule, for otherwise, Lemma 3.1 or 3.2 would be violated in a state reachable by OPT at time k. Since the order in which new jobs are assigned cannot affect expected total flow times, we may choose I_i as an assignment made by OPT at time k in *every* sample schedule starting in state σ_0. For the case (if it can occur) when OPT makes two assignments at time k, I_i is taken to be the second assignment.

Let $j=f(\sigma_0)$ be OPT's T-decision at time 0. As in Lemmas 3.1 and 3.2, we construct a policy A that simulates OPT except for an interchange of I- and T-decisions. We consider the cases $c \neq \lambda$ and $c=\lambda$ separately. Figure 3.4 illustrates the construction.

Case 1 $(c \neq \lambda)$: For A's decisions at times 0 and 1, we define

$$f^A(c, I, T)=i, \tag{3.11}$$

$$f^A(c+1, I-\{i\}, T)=f^A(c+1, I-\{i\}, T+\{i\})=j, \tag{3.12}$$

thus implementing the desired interchange, independently of whether X_i is short or long. Now with one exception, let $\sigma'=(c', I', T')$ be any state reachable by OPT at some time $t \in\{k-c+1, \ldots, k\}$. The exception occurs when $c'=\lambda$, and hence $t=k$; in this case, σ' can only be the first decision state at time k. (Recall that I_i is assigned by OPT in the second state at time k, if both machines become available at time k.)

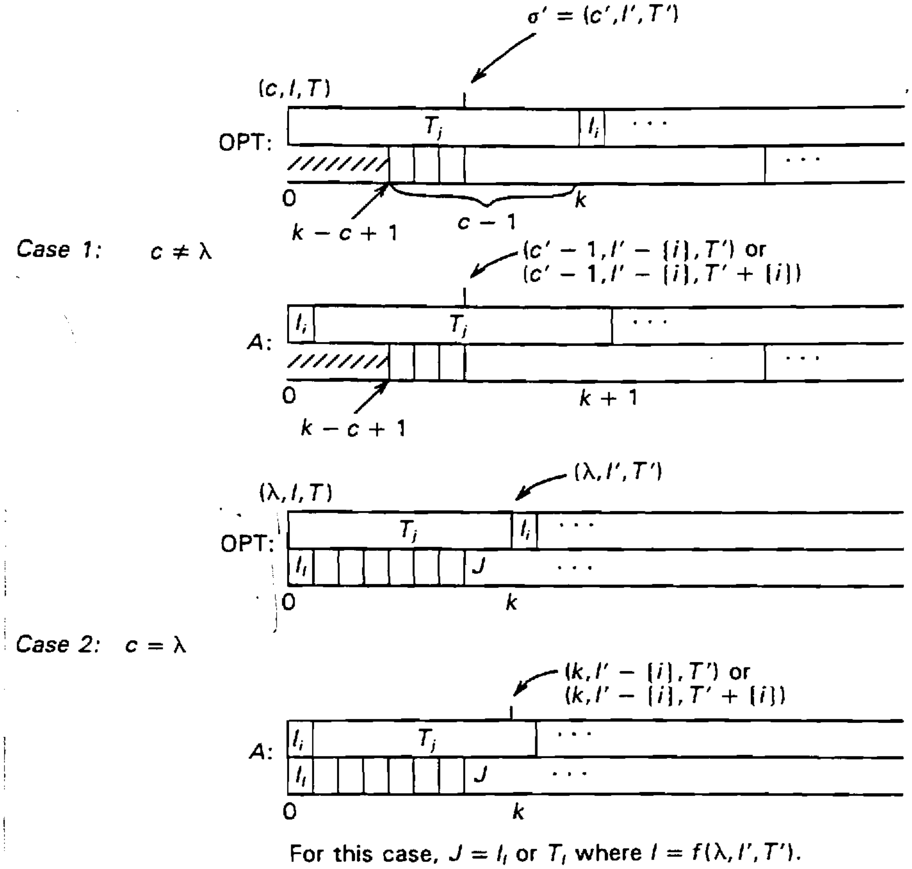


FIGURE 3.4. Pairs of schedules for A and OPT under the contradiction of Theorem 3.1.

Note that OPT must reach σ' from σ_0 by $|I| - |I'|$ consecutive I-decisions, where $|I| - |I'| = t - (k - c + 1) \le c - 1$. We define for each such state σ'

$$f^A(c' - 1, I' - \{i\}, T') = f^A(c' - 1, I' - \{i\}, T' + \{i\}) = f(\sigma'). \quad \textbf{(3.13)}$$

This definition reflects the fact that if A and OPT are to make the same decision at time t on the machine not running T_j, then the state under A must indicate that one fewer unit of time has elapsed for T_j, that I_i has already been run, and that there is an additional tail job (T_i) if running I_i revealed X_i as a long job. Note that if $|I| - |I'| = c - 1$ (i.e., $t = k$), then $c' = \lambda$ and $c' - 1 = k$;

in this case, $f(\lambda, I', T') \neq i$, since by assumption I_i is the second assignment at time k.

In states σ reachable from σ_0, other than those in Eqs. (3.11)-(3.13), we define

$$f^A(\sigma) = f(\sigma). \tag{3.14}$$

It is clear that, apart from the interchange in Eqs. (3.11) and (3.12), A simulates OPT. In particular, the decision state reached by A at the completion of T_j is the same as the decision state reached by OPT at the completion of I_i, assuming the same sample of new jobs indexed by I. As in Lemmas 3.1 and 3.2, we obtain the contradiction in Eq. (3.5).

Case 2 ($c = \lambda$): By Lemma 3.1, OPT's two assignments at time 0 are T_j and some new job, in that order. Let $l = f(1, I, T - \{j\}) \in I$ index the new job assigned at time 0. For A's decisions at time 0, we define

$$f^A(\lambda, I, T) = i, \qquad f^A(0, I - \{i\}, T\} = l.$$

For A's first decision at time 1, we define for each $Z = \phi, \{i\}, \{l\}, \{i,l\}$, the T-decision

$$f^A(\lambda, I - \{i,l\}, T + Z) = j.$$

For A's second decision at time 1, we define

$$\begin{aligned} f^A(1, I - \{i,l\}, T - \{j\}) &= f^A(1, I - \{i,l\}, T - \{j\} + \{i\}), \\ &= f(2, I - \{l\}, T - \{j\}), \end{aligned}$$

for the case when X_l is short, and

$$\begin{aligned} f^A(1, I - \{i,l\}, T - \{j\} + \{l\}) &= f^A(1, I - \{i,l\}, T - \{j\} + \{i,l\}), \\ &= f(2, I - \{l\}, T - \{j\} + \{l\}), \end{aligned}$$

for the case when X_l is long. In both cases, the decision may be an I-decision or a T-decision. By inspection, it can be seen that A makes all of the assignments made by OPT at times 0 and 1, for any given sample of new jobs. Policy A also assigns I_i but delays the start of T_j by one time unit. The remainder of the construction follows that of case 1. Again, A is uniquely defined and A and OPT converge to the same state after A completes T_j and OPT completes I_i. We again obtain the contradiction in Eq. (3.5).

The next theorem further restricts the structure of OPT schedules when $p > 1/k$. With this added restriction a strong turnpike property can be proved, and a simple bound can be derived on the expected increase in total flow time incurred by PS.

THEOREM 3.2: *If $p > 1/k$, then in any optimal decision sequence starting in state σ, there are at most k T-decisions while at least one new job is waiting. In addition,*

$$E[\mathrm{PS}(\sigma) - \mathrm{OPT}(\sigma)] \leq k(k+1)/2. \tag{3.15}$$

PROOF: By Theorem 3.1, an OPT schedule must begin with a region R_1 containing at least $n - k$ new-job assignments before the first tail job is started, at t_1 say. A region R_2 then extends from t_1 to the time t_2, when the last new job completes its initial time unit (see Figure 3.1). We now show that at most k tail jobs are started in R_2.

Let $\tau_1, \tau_2, \ldots, \tau_l$ be the starting times of tail jobs in R_2 ($\tau_1 = t_1, \tau_l < t_2$), in the order they are assigned. We assume $l \geq 2$, since the first part of the theorem follows trivially from $k \geq 2$ otherwise. We claim that

$$\tau_{j-1} < \tau_j < \tau_{j-1} + k, \qquad 2 \leq j \leq l. \tag{3.16}$$

Lemma 3.1 establishes the first inequality $\tau_{j-1} < \tau_j$. For the second inequality, suppose $\tau_j \geq \tau_{j-1} + k$. Then in the state (c, I, T) where the $(j-1)$st tail job is assigned, the inequality $|I| \geq c - 1$ must hold (see Figure 3.5). But if $|I| = c - 1$, then τ_{j-1} must be the time of the last T-decision in R_2, and if $|I| > c - 1$, then Theorem 3.1 is violated. Thus, Eq. (3.16) holds.

Now if $l > 2$, then Eq. (3.16) along with Lemma 3.2 implies that at least one I-decision must be made between the $(j-1)$st and jth T-decisions. Thus, on each machine in R_2, T-decisions alternate with sequences of one or more consecutive I-decisions. Since at most k new jobs start in R_2 (by Theorem 3.1 applied to the first T-decision), there can be at most k tail jobs started in R_2. This proves the first part of the theorem.

For the bound given by (Eq. (3.15), we observe that the expected flow time of short jobs according to PS is at most that according to OPT. Thus, we can obtain a crude bound by calculating a deterministic worst-case increase in the finishing times of long jobs.

As illustrated in Figure 3.1, consider a sample schedule under OPT and the corresponding schedule under PS. It is easy to verify that we can transform the OPT schedule in R_2 to the corresponding PS schedule in $[t_1, t_2]$ so that at

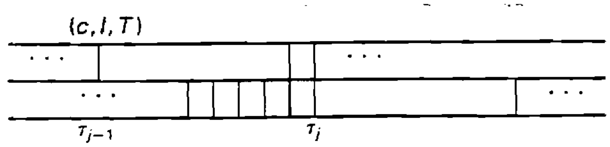


FIGURE 3.5. Example for Theorem 3.2.

most $\left\lceil \frac{l}{2} \right\rceil$ is added to the finishing time of each tail job in R_2. Since $l \le k$, we have an increase of at most $k^2/2$ in the flow times of tail jobs started in R_2.

Let $m \ge 1$ tail jobs be started in R_3 according to OPT. The starting time of the first tail job started in R_3 is t_2: let t_2' be the starting time of the second, if any. The above transformation from OPT to PS in general changes these starting times to t_3 and t_3', where $t_3 + t_3' = t_2 + t_2'$ (one starting time is shifted left and the other shifted right by an equal amount) and $t_3' - t_3 < k$. This creates no change in the total flow time of tail jobs starting in R_3 if m is even, and a change of at most $k/2$ if m is odd. The increase in the total flow time of long jobs is therefore no greater than $k^2/2 + k/2 = k(k+1)/2$. ■

In this section, our results have been based on the assumption that $p > 1/k$. This has been for convenience, since we could have also allowed equality, i.e., $p \ge 1/k$. Lemmas 3.1 and 3.2 and Theorems 3.1 and 3.2 would then have been modified to state that *there exists* an optimal policy, rather than *there must be* an optimal policy with the claimed properties. With these changes, we can conclude that PS is a (strongly) optimal turnpike policy for all $p \ge 1/k$.

4. THE OPT AND NS POLICIES FOR $p < 1/k$

The asymptotics in this section are obtained from direct calculations of expected NS flow times. Accordingly, we begin with derivations of the appropriate formulas and a study of their properties.

Given an initial state $\sigma = (c, I, T)$, the expected total flow time is a function only of c, $n = |I|$, and $r = |T|$. In this section, we exploit this fact and use (c,n,r) as the decision state. Policy A is now defined by a simpler mapping f^A, where $f^A(c,n,r)$ takes on one of two values, one calling for a new-job assignment and the other for a tail-job assignment. The terminology of i- and T-decisions is used as before. For the purposes of this section, any order of selection from I and T may be assumed. Initial states (c,n,r) are general with $n \ge 0$, $r \ge 0$ $(n + r > 0)$, and $c = \lambda$ or $0 \le c \le k$. We call (n,r) the *backlog* of state (c,n,r). The *level* of (c,n,r) is simply n, the number of new jobs.

Observe that states in which NS assigns a new job must have the form $(c,n,0)$, i.e., no tail jobs are waiting to be assigned. Also, after the first new job has been assigned, the only states reachable by NS have the form $(c,n,0)$, $(c,n,1)$, or $(\lambda,n,2)$.

Under NS, let C_i, $i \ge 0$, denote the elapsed-time random variable in the state where the $(i+1)$st new job is assigned. C_0 corresponds to the initial state if there are no waiting tail jobs in the initial state. By the above observations it is easily verified that, for any given C_0, $\{C_i;\ 0 \le i \le n-1\}$ is a finite, irreducible Markov chain. However, the analysis of NS is more natural in terms of the *remaining* times on the occupied machine, when new jobs are assigned. Thus, we define the sequence of *resume delays* $D_0, D_1, \ldots, D_{n-1}$, where $D_i =$

$k - C_i + 1$ if $C_i \neq 0$, and $D_i = 1$ or $k + 1$ with probabilities p and $1 - p$, respectively, if $C_i = 0$. After calculating the *conditional* expected flow times

$$E\text{NS}(n,r|d) \equiv E\text{NS}(n,r|D_0 = d),$$

we then obtain

$$\begin{aligned} E\text{NS}(c,n,r) &= E\text{NS}(n,r|k - c + 1), \qquad c \neq 0, \\ E\text{NS}(0,n,r) &= pE\text{NS}(n,r|1) + (1 - p)E\text{NS}(n,r|k + 1). \end{aligned} \tag{4.1}$$

Before giving these calculations, we discuss briefly the properties of the chain $\{D_i\}$. With D_0 given, the chain is defined by the recurrence

$$D_{i+1} = |D_i - X_{i+1}|, \qquad i = 0,1,2,\ldots, \tag{4.2}$$

and is therefore a finite, irreducible Markov chain on the set $\{0,1,\ldots,k+1\}$. Note that if k is odd, the chain is ergodic. If k is even, then the X_i's must all be odd (each is 1 or $k + 1$), and hence they create an alternating sequence of parities in the D_i's. Thus, the chain has period 2. A brief analysis of the limiting distributions is given in the Appendix. This analysis shows that if

$$\alpha = \frac{1}{2}\left[\frac{1}{\mu} + \mu p(1 - p)k^2\right] = \frac{EX^2}{2EX}, \tag{4.3}$$

where $1/\mu \equiv EX$, then as $i \to \infty$,

$$ED_i \to \alpha, \qquad k \text{ odd},$$

$$ED_i + \xi_{i+d}\frac{\mu}{2} \to \alpha, \qquad k \text{ even},$$

where ξ_j is the parity function, $\xi_j = +1$, j even, and $\xi_j = -1$, j odd. As expected, α is simply the mean forward-recurrence time of a renewal process with intervals between renewals having the distribution of job running times.

In calculating expected flow times, we condition on an initial resume delay d on the occupied machine and consider first an initial backlog of $(n,0)$. It is easy to see that the ith job, $i \geq 1$, begins at time $\frac{1}{2}\left[d + \sum_{1\leq j\leq i-1} X_j\right] - D_{i-1}/2$ with the expected value $\frac{1}{2}[d + (i-1)/\mu] - ED_{i-1}/2$. We add $1/\mu$ to the latter expression to obtain the expected finishing time of the ith job. Thus, for the expected total flow time, we have

$$\begin{aligned} E\text{NS}(n,0|d) &\equiv E\text{NS}(n|d) \\ &= \sum_{1\leq i\leq n}\left[\frac{d + (i-1)/\mu - ED_{i-1}}{2} + \frac{1}{\mu}\right]. \end{aligned} \tag{4.4}$$

In the summation of Eq. (4.4), it is convenient to add and subtract $n\alpha/2$ and put the result in the form

$$E\mathrm{NS}(n|d) = \frac{n(n+3)}{4\mu} + \frac{n}{2}(d-\alpha) - \frac{1}{2}h_n(d), \tag{4.5}$$

where

$$h_n(d) = \sum_{0\le i\le n-1} E(D_i - \alpha | D_0 = d). \tag{4.6}$$

Before analyzing the function $h_n(d)$, we extend Eq. (4.5) to initial states with tail-job backlogs $r \ge 1$. By the NS rule these tail jobs are run first. At the point when the tail-job backlog first reduces to 0, the initial resume delay D_0 begins. Clearly, D_0 is a deterministic function of the initial delay d. For $0 \le d \le k$, $D_0 = d$ if $r \le 1$ is even, and $D_0 = k - d$ if r is odd. If $d = k + 1$, then $D_0 = k - 1$ if $r \ge 1$ is even, and $D_0 = 1$ if r is odd. We denote this relation by $D_0 = \delta(d,r)$, and define $\delta(d,0) = d$. The dependence on d and r will often be suppressed when obvious in context.

A calculation now shows that for $r \ge 1$ and $0 \le d \le k$,

$$\begin{aligned} E\mathrm{NS}(n,r|d) &= \frac{r}{4}[2d + (r+2)k] + \frac{1}{4}(\delta - d) \\ &\quad + E\mathrm{NS}(n|\delta) + \frac{n}{2}[rk - (\delta - d)], \end{aligned} \tag{4.7}$$

where the first two terms on the right-hand side give the expected flow time of the first r tail jobs, the third term is the expected flow time of the n new jobs, assuming that the first new-job assignment is made at time 0, and the last term corrects the preceding term by adding n times the instant, $[rk - (\delta - d)]/2$, when the first new job is actually assigned. An expression for $d = k + 1$, $r \ge 1$ can be found from Eq. (4.7) and

$$E\mathrm{NS}(n,r|k+1) = (n+r)k + E\mathrm{NS}(n,r-1|1). \tag{4.8}$$

We return now to an analysis of the asymptotics of the expected value $h_n(d)$ in Eq. (4.6). Define

$$h(d) = \frac{\mu}{d}\left(d^2 - \frac{1}{6}\right) - \frac{\mu^2}{6}EX^3, \qquad 0 \le d \le k+1, \tag{4.9}$$

where $EX^3 = p + (1-p)(k+1)^3$. In the Appendix, we prove the following result.

THEOREM 4.1: *If k is odd, then $h_n(d) \to h(d)$ as $n \to \infty$. If k is even, then $h_{2n}(d) \to h(d) + \xi_d\frac{\mu}{4}$ and $h_{2n+1}(d) \to h(d) - \xi_d\frac{\mu}{4}$ as $n \to \infty$.*

The maximum of Eq. (4.9) is obtained at $d = k + 1$. For $p < 1/k$, a calculation yields the bound

$$\max_{0 \le d \le k+1} |h(d)| \le \frac{k+2}{2}. \tag{4.10}$$

Next, we prove two lemmas from which the asymptotic optimality of NS will follow easily. Lemma 4.1 eases the calculations in the proof of Lemma 4.2; this will be further explained just prior to the statement of Lemma 4.2.

For the remaining results of this section, we define $D(\sigma)$, the *final delay* produced by NS starting in state σ, as the delay on the occupied machine until the end of the schedule, measured from the time when the backlog first reaches (0,0); in other words, $D(\sigma)$ is simply the difference in machine finishing times. Note that final delays are distinguished from resume delays by the absence of a subscript. As before, $D(n,r|d)$ and $D(n|d) \equiv D(n,0|d)$ are conditioned on an initial delay of d time units on the occupied machine.

LEMMA 4.1: *Let $h(D(n|d))$ be the random variable given by Eq. (4.9) with d replaced by $D(n|d)$. We have*

$$h_n(d) + Eh(D(n|d)) = h(d). \tag{4.11}$$

PROOF: We write for any $s \ge 1$

$$h_{n+s}(d) = E\left[\sum_{j=0}^{n+s-1} (D_j - \alpha)|D_0 = d\right],$$

$$= h_n(d) + E\left[\sum_{j=n}^{n+s-1} (D_j - \alpha)|D_n(n+s|d)\right].$$

But the second term on the right of this last expression is simply $Eh_s(D_n(n + s|d))$. Since $D_n(n + s|d)$ and $D(n|d)$ are equal in distribution for all $s \ge 1$, we have

$$h_{n+s}(d) = h_n(d) + Eh_s(D(n|d)).$$

Taking the limit $s \to \infty$ through the even integers and applying Theorem 4.1 now yields Eq. (4.11). ∎

The notation $D_n^A(\sigma)$ and $D^A(\sigma)$ extends our definitions of resume and final delays to an arbitrary policy A. For any policy A starting in state σ, we define the *modified flow time*

$$A^*(\sigma) = A(\sigma) - \frac{1}{2} h(D^A(\sigma)), \tag{4.12}$$

By Eq. (4.10), the expected difference in $A(\sigma)$ and $A^*(\sigma)$ is bounded by

$$|EA^*(\sigma) - EA(\sigma)| < (k+2)/4. \tag{4.13}$$

The usefulness of the new cost function stems from Eq. (4.13) and the fact, implied by Lemma 4.2 below, that its expected value is minimized by NS.

From Eq. (4.11), we note that $Eh(D^{NS}(n,r|d)) \to 0$ as $n \to \infty$. Thus, $E\text{NS}^*(n,r|d) \to E\text{NS}(n,r|d)$ as $n \to \infty$ for any fixed r and d. The expected value of the flow time modification and the relation of Eq. (4.11) also have important consequences in the calculation of $E\text{NS}^*(n,r|d)$, which we now describe.

Observe that $E\text{NS}(n|d)$ fails to be an explicit function of n because of the term $-\frac{1}{2}h_n(d)$ in Eq. (4.5). However, if we further subtract $\frac{1}{2}Eh(D(n|d))$, then the two terms combine into $-\frac{1}{2}[h_n(d) + Eh(D(n|d))]$, which by Eq. (4.11) is an explicit function of n. Since the subtraction of $\frac{1}{2}Eh(D(n|d))$ from $E\text{NS}(n|d)$ gives us $E\text{NS}^*(n|d)$ by definition (see Eq. (4.12)), we therefore obtain $E\text{NS}^*(n|d)$ as an explicit function of n, i.e., the right-hand side of Eq. (4.5) with $h_n(d)$ replaced by $h(d)$.

A similar statement applies to Eqs. (4.7) and (4.8) for $r > 0$, where the term $-\frac{1}{2}h_n(\delta)$ appears implicitly. Here, subtraction of

$$\frac{1}{2}\,Eh(D(n,r|d)) = \frac{1}{2}\,Eh(D(n|\delta))$$

produces explicit functions of n in which $h_n(\delta)$ is replaced by $h(\delta)$.

Under a given policy A, the state $(c,n,r), n,r \geq 1$, is called a *preemption state* if a new job is assigned in that state. The corresponding decision, $f^A(c,n,r)$, is called a preemption. The following lemma proves that NS is optimal for the expected modified flow time, as well as supplying the bound needed for Theorem 4.2.

LEMMA 4.2: *If $p < 1/k$, then for any policy A, we have*

$$E[A^*(\sigma) - \text{NS}^*(\sigma)] \geq \rho^A(\sigma)(1 - pk)\mu, \tag{4.14}$$

where $\rho^A(\sigma)$ is the expected number of preemptions made by A starting in state σ.

PROOF: Based on a given policy A and an initial state $\sigma = (c,n,r)$, we define a sequence of policies $A_1, A_2, \ldots, A_{n+1}$, where $A_1 \equiv A$, $A_{n+1} \equiv \text{NS}$, and A_j, $2 \leq j \leq n$, is identical to A in states at level j or greater, but is identical to NS in states at levels less than j. We write

$$E[A^*(\sigma) - \text{NS}^*(\sigma)] = \sum_{j=1}^{n} E[A_j^*(\sigma) - A_{j+1}^*(\sigma)]. \tag{4.15}$$

To obtain a lower bound on $E[A_j^*(\sigma) - A_{j+1}^*(\sigma)]$, suppose $\sigma_j = (c_j,j,r_j)$ is a preemption state under A, and hence A_j. By definition of A_j and A_{j+1}, policy A_j starts with a preemption in (c_j,j,r_j), but is otherwise non-preemptive, and policy A_{j+1} is simply NS in state σ_j; i.e., $A_{j+1}^*(\sigma_j) = \text{NS}^*(\sigma_j)$.

For the case $1 \le c_j \le k$, and hence $1 \le d = k - c_j + 1 \le k$, we write for A_j

$$EA_j^*(\sigma_j) = EA_j(\sigma_j) - \frac{1}{2} Eh(D^{A_j}(\sigma_j)).$$

Since A_j begins with a preemption, we can substitute for $EA_j(\sigma_j)$ as follows:

$$EA_j^*(\sigma_j) = j + r_j + p\text{NS}(j-1, r_j | d-1) + (1-p)E\text{NS}(j-1, r_j+1 | d-1) - \frac{1}{2} Eh(D^{A_j}(\sigma_j)). \quad \textbf{(4.16)}$$

Next, by definition of A_j, we can substitute

$$Eh(D^{A_j}(\sigma_j)) = pEh(D(j-1, r_j | d-1)) + (1-p)Eh(D(j-1, r_j+1 | d-1))$$

← Author: missing parentheses inserted correctly?

into Eq. (4.16), and then substract $E\text{NS}^*(j, r_j | d)$ to obtain

$$E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] = j + r_j + pE\text{NS}^*(j-1, r_j | d-1) + (1-p)E\text{NS}^*(j-1, r_j+1 | d-1) - E\text{NS}^*(j, r_j | d). \quad \textbf{(4.17)}$$

Similarly, for $c_j = \lambda$, we obtain

$$E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] = p[1 + E\text{NS}^*(j-1, r_j | 1)] + (1-p)[k+1+E\text{NS}^*(j-1, r_j | k+1)] - E\text{NS}^*(j, r_j | 0), \quad \textbf{(4.18)}$$

and for $c_j = 0$, we have by Eq. (4.1),

$$\begin{aligned} E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] &= p[j + r_j + pE\text{NS}^*(j-1, r_j | 0) \\ &\quad + (1-p)E\text{NS}^*(j-1, r_j+1 | 0)] \\ &\quad + (1-p)[j + r_j + pE\text{NS}^*(j-1, r_j | k) \qquad \textbf{(4.19)} \\ &\quad + (1-p)E\text{NS}(j-1, r_j+1 | k)] \\ &\quad - [pE\text{NS}^*(j, r_j | 1) + (1-p)E\text{NS}^*(j, r_j | k+1)]. \end{aligned}$$

Using Eqs. (4.5), (4.7), and (4.8) with $h_n(\delta)$ replaced by $h(\delta)$, we can evaluate the right-hand sides of Eqs. (4.17)–(4.19) as explicit functions of the parameters and obtain the uniform bound

$$E[A_j^*(\sigma_j) - A_{j+1}^*(\sigma_j)] \ge \mu(1 - pk). \quad \textbf{(4.20)}$$

The calculations, although lengthy, are routine. In the Appendix, we show how the calculations can be done more simply by exploiting the fact that A_j and A_{j+1} differ in only one decision.

Now let $q_j^A(\sigma)$ denote the probability that A, starting in state σ, reaches one of the preemption states (c_j, j, r_j) at level j. Then

$$E[A_j^*(\sigma) - A_{j+1}^*(\sigma)] \geq q_j^A(\sigma)\mu(1-pk), \qquad 1 \leq j \leq n.$$

and

$$E[A^*(\sigma) - \mathrm{NS}^*(\sigma)] \geq \mu(1-pk)\sum_{j=1}^{n} q_j^A(\sigma). \tag{4.21}$$

Since $\rho^A(\sigma) = \sum_{j=1}^{n} q_j^A(\sigma)$, Eq. (4.14) follows. ■

It is now easy to prove the main result of this section: Both the expected added cost of NS relative to OPT and the expected number of preemptions made by OPT are bounded by constants independent of the initial state.

THEOREM 4.2: *If $p < 1/k$, then for any initial state σ,*

$$E[\mathrm{NS}(\sigma) - \mathrm{OPT}(\sigma)] \leq (k+2)/2, \tag{4.22}$$

and

$$\rho^{\mathrm{OPT}}(\sigma) \leq \frac{k+2}{2\mu(1-pk)}. \tag{4.23}$$

PROOF: By Eqs. (4.10) and (4.12), we have

$$|E\mathrm{NS}^*(\sigma) - E\mathrm{NS}(\sigma)| \leq (k+2)/4, \tag{4.24}$$

$$|E\mathrm{OPT}^*(\sigma) - E\mathrm{OPT}(\sigma)| \leq (k+2)/4. \tag{4.25}$$

From Eq. (4.24) and Lemma 4.2, we obtain

$$\begin{aligned} E\mathrm{NS}(\sigma) &\leq E\mathrm{NS}^*(\sigma) + (k+2)/4, \\ &\leq E\mathrm{OPT}^*(\sigma) + (k+2)/4. \end{aligned}$$

Substitution of $E\mathrm{OPT}^*(\sigma) \leq E\mathrm{OPT}(\sigma) + (k+2)/4$ yields Eq. (4.22).

As above, we can apply Eq. (4.25), the inequality $E\mathrm{OPT}(\sigma) \leq E\mathrm{NS}(\sigma)$, and Eq. (4.24), in that order, to obtain

$$E\mathrm{OPT}^*(\sigma) - E\mathrm{NS}^*(\sigma) \leq (k+2)/2. \tag{4.26}$$

By Lemma 4.2, we have

$$E\mathrm{OPT}^*(\sigma) - E\mathrm{NS}^*(\sigma) \geq \rho^{\mathrm{OPT}}(\sigma)\mu(1-pk). \tag{4.27}$$

The bound given by Eq. (4.23) follows directly from Eqs. (4.26) and (4.27). ■

5. CONCLUSIONS

A problem, which is arguably the simplest multimachine flow-time scheduling problem with job running times not having a monotone hazard-rate distribution, has been shown to have a surprisingly rich structure. In particular, it appears that the optimal policy cannot be determined by a "ranking function." Nevertheless, the SERPT rule was proved to be asymptotically optimal in a strong sense: the expected number of OPT decisions that are not SERPT decisions is bounded by a constant independent of the initial state. An even stronger turnpike optimality was proved for the case $p \geq 1/k$. Equally satisfying was the result (see Eqs. (3.3) and (4.22)) that SERPT yields an expected flow time which exceeds the optimal value by a bounded amount, independent of the initial state.

There are several obvious challenges for future research:

1. Prove that property (P2) of Section 1 is not restrictive. This should follow from Bellman equations.
2. Resolve the question of whether a turnpike theorem can be proved for NS as was the case for PS (see Theorem 3.2).
3. Extend the results to 3 or more machines, to the expected makespan objective function, and to two-point distributions, where the larger point is not a multiple of the smaller point.

Acknowledgment

Research of G. Weiss was supported in part by the National Science Foundation Grant ECS 8712798.

References

1. Agrawala, A., Coffman, E.G., Jr., Garey, M.R., & Tripathi, S. (1984). A stochastic optimization algorithm minimizing expected flow times on uniform processors. *IEEE Transactions on Computers* C-33: 351–356.
2. Bruno, J.L., Downey, P.J., & Frederickson, G.N. (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the Association for Computing Machines* 28: 100–113.
3. Chazan, D., Konheim, A.G., & Weiss, B. (1968). A note on time-sharing. *Journal of Combinatorial Theory* 5: 344–369.
4. Coffman, E.G., Jr., Flatto, L., Garey, M.R., & Weber, R.R. (1987). Minimizing expected makespans on uniform processor systems. *Advances in Applied Probability* 19: 177–201.
5. Feller, W. (1971). *An introduction to probability theory and its applications*, Vol. II, 2nd Edition, New York: John Wiley & Sons.
6. Gittins, J.C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B* 14: 148–177.
7. Glazebrook, K.D. (1979). Scheduling tasks with exponential service times on parallel processors. *Journal of Applied Probability* 16: 658–689.

8. Kämpke, T. (19xx) On the optimality of static priority policies in stochastic scheduling on parallel machines. *Journal of Applied Probability* (to appear).
9. Meilijson, I. & Weiss, G. (1977). Multiple feedback at a single server station. *Stochastic Processes and Their Applications* 5: 195–205.
10. Papadimitriou, C. (19xx). Games against nature. *Symposium on the Foundations of Computer Science*, IEEE Comp. Soc. Press, pp. 446–450 (to appear *J. Comp. Sys. Sci.*).
11. Ross, S. (1983). *Introduction to stochastic dynamic programming.* New York: Academic Press.
12. Sevcik, K.C. (1974). Scheduling for minimum total loss using service time distributions. *Journal of the Association for Computing Machines* 21: 66–75.
13. Weber, R.R. (1982). Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time. *Journal of Applied Probability* 19: 167–182.
14. Weber, R.R., Varaiya, P., & Walrand, J. (1986). Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flow time. *Journal of Applied Probability* 23: xx–xx.
15. Weiss, G. & Pinedo, M. (1980). Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions. *Journal of Applied Probability* 17: 187–202.

Author: Can you update Refs 8 and 10?

Please check to determine if Ref. 11 is called out in text. Appears not to be.

Please provide pg. nos. for Ref. 14.

APPENDIX

Properties of $\{D_n\}$. Because of our present need for more indices, we will depart slightly from the conventions in Section 4. We use i and j, $0 \le i, j \le k+1$, to denote delay states; and m and n are used as time parameters.

Feller [5] has analyzed the chain $D_{n+1} = |D_n - X_{n+1}|$ when the X_n's are i.i.d. with a strictly positive density on $[0,\infty]$. In our case, X_n is arithmetic, so Feller's formulas do not apply. On the other hand, $\{D_n\}$ is a finite Markov chain, so geometrically fast convergence to stationary distributions is assured. Properties of $\{D_n\}$ can be derived as follows.

The given distribution of a job's running time, X, is $Pr\{X = 1\} = p$ and $Pr\{X = k+1\} = 1-p$. The chain $\{D_n\}$ has values in $\{0,1,\ldots,k+1\}$, on the assumption that $0 \le D_0 \le k+1$. $\{D_n\}$ is easily seen to be irreducible; for k odd it is aperiodic (and hence ergodic), and for k even it has period 2. The transition matrix $\mathbf{P} = \{p_{ij}\}$ is given by

$$p_{ij} = \begin{cases} p & i=0, j=1 \text{ or } i=1,\ldots,k+1,\ i \ne \frac{k}{2}+1,\ j=i-1 \\ 1-p & i=0,\ldots,k+1,\ i\ne \frac{k}{2}+1,\ j=k+1-i \\ 1 & k \text{ even},\ i=\frac{k}{2}+1,\ j=\frac{k}{2} \\ 0 & \text{otherwise.} \end{cases} \tag{A1}$$

By inspection, a solution $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{k+1})$ to $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ is given by

$$\pi_i = \begin{cases} \mu/2 & i=0, \\ (2-p)\mu/2 & i=1, \\ (1-p)\mu & 2 \le i \le k, \\ (1-p)\mu/2 & i=k+1, \end{cases} \tag{A2}$$

where $1/\mu = EX = k+1-kp$ is the mean job-running time. For k odd, the distribution of D_n converges geometrically fast to $\boldsymbol{\pi}$; for k even, the distribution converges geometrically fast to $\boldsymbol{\pi}^{(e)}$, with $\pi_i^{(e)} = 2\pi_i \dfrac{1+\xi_i}{2}$, and to $\boldsymbol{\pi}^{(o)}$, with $\pi_i^{(o)} = 2\pi_i \dfrac{1-\xi_i}{2}$, according as D_n is even or odd, respectively. (Recall that ξ_i is the parity function, $\xi_i = +1$ for m even, and $\xi_i = -1$ for m odd.) By direct calculation, the first moment of $\boldsymbol{\pi}$ is α, as written in Eq. (4.3). The first moments of $\boldsymbol{\pi}^{(e)}$ and $\boldsymbol{\pi}^{(o)}$ work out to be $\alpha - \mu/2$ and $\alpha + \mu/2$, respectively. Hence, for k odd, $ED_n \to \alpha$ as $n \to \infty$, while for k even, $E(D_n | D_0 = d) + \xi_{n+d}\dfrac{\mu}{2} \to \alpha$ as $n \to \infty$; again, these convergences are geometrically fast.

PROOF OF THEOREM 4.1. From the definition of $h_n(d)$ (see Eq. (4.6)), we obtain the recurrence

$$h_n(i) = i - \alpha + \sum_{j=0}^{k+1} p_{ij} h_{n-1}(j). \tag{A3}$$

First, consider k odd. The geometric convergence of ED_n to α as $n \to \infty$ implies a similar convergence of $\sum_{n=0}^{\infty} E(D_n - \alpha | D_0 = d)$ to a value, say $h(d)$. Letting $n \to \infty$ in Eq. (A3), we obtain the equations

$$h(i) = i - \alpha + \sum_{j=0}^{k+1} p_{ij} h(j), \qquad i = 0,1,\ldots,k+1. \tag{A4}$$

To solve Eq. (A4), we define the differences $\Delta(i) = h(i+1) - h(i)$. Exploiting the form of p_{ij} in Eq. (A1), we obtain

$$\Delta(0) = 1 - p\Delta(0) - (1-p)\Delta(k),$$
$$\Delta(i) = 1 + p\Delta(i-1) - (1-p)\Delta(k-i), \qquad 1 \le i \le k,$$

which is solved by

$$\Delta(i) = h(i+1) - h(i) = \frac{\mu}{2}(2i+1), \qquad 0 \le i \le k.$$

It follows that $h(i) - h(0) = \frac{\mu}{2} i^2$, $0 \le i \le k+1$. Also, $\sum_{i=0}^{k+1} \pi_i h_n(i) = 0$ for all n, so that $\sum_{i=0}^{k+1} \pi_i h(i) = 0$. Thus,

$$-h(0) = \sum_{i=0}^{k+1} \pi_i [h(i) - h(0)] = \frac{\mu}{2} \sum_{i=0}^{k+1} \pi_i i^2,$$
$$= \frac{\mu}{12} + \frac{\mu^2}{6} EX^3.$$

Theorem 4.1 follows for k odd.

For k even, $E(D_n | D_0 = d) + \xi_{n+d} \frac{\mu}{d} \to \alpha$ geometrically fast, and therefore

$$h_n(d) + \frac{\mu}{2} \sum_{0 \le m \le n-1} \xi_{m+d}$$

converges geometrically fast, to $g(d)$ say. The sum is 0 for n even, and ξ_d for n odd. Then as $n \to \infty$.

$$h_{2n}(d) \to g(d),$$
$$h_{2n+1}(d) \to g(d) - \frac{\mu}{2} \xi_d.$$

Letting $n \to \infty$ in the recurrence Eq. (A3), and noting that $p_{ij} = 0$ for $\xi_i = \xi_j$, we obtain the equations

$$\left[g(i) - \frac{\mu}{4}\xi_i\right] = (i-\alpha) + \sum_{j=0}^{k+1} p_{ij}\left[g(j) - \frac{\mu}{4}\xi_j\right], \qquad 0 \le i \le k+1,$$

from which, similar to the analysis for k even, we arrive at

$$g(i) - g(0) = \begin{cases} i^2 \dfrac{\mu}{2} & i \text{ even}, \\ (i^2 - 1)\dfrac{\mu}{2} & i \text{ odd}. \end{cases}$$

Observing that $\sum_{i=0}^{k+1} \pi_i^{(e)} h_{2n}(i) = 0$ for all n, we have $\sum_{i=0}^{k+1} \pi_i^{(e)} g(i) = 0$, and hence

$$-g(0) = \sum_{i=0}^{k+1} \pi_i^{(e)}[g(i) - g(0)] = \sum_{i=0}^{k+1} \pi_i^{(e)} i^2 \frac{\mu}{2},$$

$$= \frac{k(k+1)(k+2)}{6}\mu^2 = \frac{\mu^2}{6}EX^3 - \frac{\mu}{6}.$$

On substitution, we finally obtain $g(i) = h(i) + \xi_i \frac{\mu}{4}$ and, as $n \to \infty$,

$$h_{2n}(d) \to h(d) + \xi_d \frac{\mu}{4},$$

$$h_{2n+1}(d) \to h(d) - \xi_d \frac{\mu}{4}. \qquad \blacksquare$$

Proof of Eq. (4.20) for $1 \le c_j \le k$. We need to prove that

$$\Delta = E[A_j^*(n,r|d) - \text{NS}^*(n,r|d)] \ge \mu(1 - pk), \qquad 1 \le d \le k.$$

The approach below, along with Eq. (4.1), is easily adapted to a proof of Eq. (4.20) for $c_j = \lambda$ and $c_j = 0$; the details are left to the interested reader. Accompanying the results below are figures describing schedules up through the first r tail-job assignments and the first new-job assignment. The dashed tail job is absent with probability p, and present with probability $1 - p$.

Case 1 $(1 \le d < k)$: For r even, inspection of Figure A1 shows that

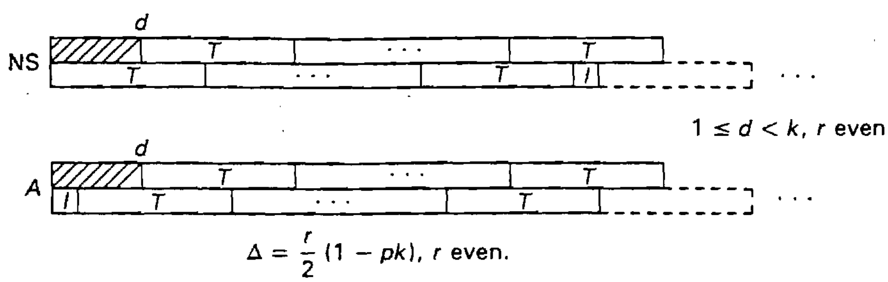
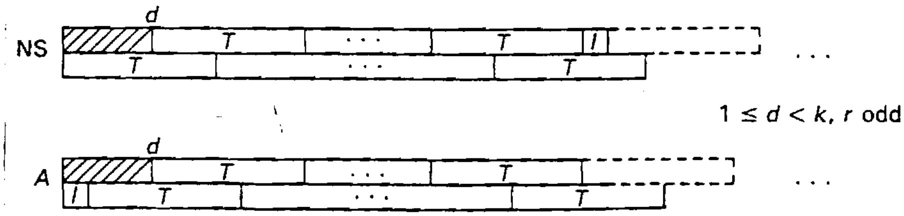


Figure A.1.

For r odd, we obtain with the help of Figure A2,



$$\Delta = \frac{r+1}{2} - \left(\frac{r-1}{2}k + d\right)p - (1-p)$$
$$- p\,\frac{h(k-d+1) - h(k-d-1)}{2} + (1-p)\,\frac{h(d+1) - h(d-1)}{2}.$$

Figure A.2.

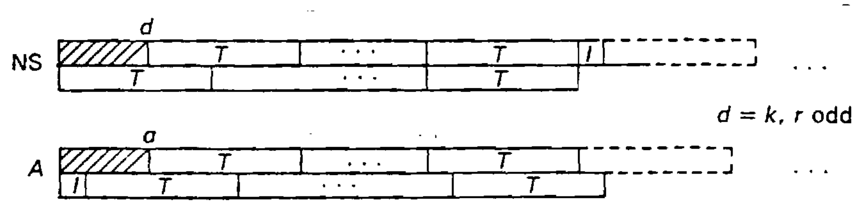
The last two terms express the difference in the flow time of the last $n-1$ complete jobs; the running of these jobs starts on the average at the same time under both policies. Note that $[h(i+1) - h(i-1)]/2 = \mu i$ and $(1 - \mu k) = \mu(1 - pk)$, so that

$$\Delta = (1-pk)\frac{r-1}{2} - pd + p + \mu d - \mu pk,$$
$$= (1-pk)\left[\frac{r-1}{2} + \mu(p + (1-p)d)\right].$$

Case 2 ($d = k$): For r even, we obtain the same result as in Case 1, namely,

$$\Delta = \frac{r}{2}(1-pk), \qquad r \text{ even}.$$

For r odd, Figure A3 shows that



$$\Delta = \frac{r+1}{2}(1-pk) - (1-p) + (1-p)\,\frac{h(k+1) - h(k-1)}{2},$$
$$= (1-pk)\left[\frac{r+1}{2} - (1-p)\mu\right],$$
$$= (1-pk)\left[\frac{r-1}{2} + \mu(p + (1-p)k)\right], \quad r \text{ odd}.$$

Figure A.3.

Accumulating the results above, we seen that Eq. (4.20) holds for $1 \le c_j \le k$.

Industrial and Systems Engineering
Report Series No. J-88-3
March, 1988

APPROXIMATION RESULTS IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

*This research was supported in part by NSF grant ECS-8712798.

APPROXIMATION RESULTS IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

Abstract

We consider scheduling a batch of jobs with stochastic processing times on parallel machines. We derive various new formulae for the expected flowtime and weighted flowtime under general scheduling rules. Smith's Rule, which orders job starts by decreasing ratio of weight to expected processing time provides a natural heuristic for this problem. We obtain an O(1) bound on the worst case difference between the expected weighted flowtime under Smith's Rule and under an Optimal policy.

*This research was supported in part by NSF grant ECS-8712798.

Approximation Results in Parallel Machines Stochastic Scheduling

Gideon Weiss
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Introduction:

One of the simplest yet most useful results in scheduling theory is that flowtime (sum of all the waiting and processing times of all the jobs) is minimized by the SPT (Shortest Processing Time First) rule. This result holds for a single processor (Smith, 1956) as well as for parallel processors (McNaughton, 1959). For the often more applicable weighted flowtime objective function (where the waiting and processing times are weighted by different cost per unit time for each job) the optimal sequence of jobs on a single processor is in decreasing order of weight to processing time ratios (Smith, 1956), the so called "Smith's Rule". Minimization of weighted flowtime on several parallel machines is, however, an NP-hard combinatorial optimization problem for any fixed number of machines greater than 1 (Garey and Johnson 1979, Lenstra, Rinnooy-Kan and Brucker 1977). Scheduling the starting times of the jobs according to Smith's rule provides a suboptimal heuristic for this problem, whose worst case performance is 1.2071 times the optimal value (Weiss et al. 1987).

In the present paper we examine a stochastic version of these problems, where we assume that the processing requirements of the jobs are not known in advance but are drawn from some known probability distributions. The single processor results generalize easily to stochastic processing times: SEPT (shortest expected processing time first) and Smith's Rule (decreasing order of weight to expected processing time ratio) respectively minimize the expected flowtime and the expected weighted flowtime.

Results for parallel machines are much more complex. SEPT remains optimal for flowtime in a wide range of problems, though the proofs are no longer elementary. SEPT is optimal when job processing times are exponentially distributed (Bruno, Downey and Frederickson 1981, Glazebrook 1979, Pinedo and Weiss 1979, Weiss and Pinedo 1980). It is also optimal when the job processing time distributions are all of them tails of a single IHR (increasing hazard rate) distribution (Weber 1982). Both these results are special cases of the remarkable result of Weber, Varaiya and Walrand (1986) that SEPT is optimal whenever the processing time distributions of all the jobs are stochastically comparable in pairs. That result seems the most general possible. It does not extend to weaker comparison conditions, and SEPT fails to be optimal in general (see Pinedo and Weiss 1987 for counter examples). Very little can be said on optimality of Smith's Rule for parallel machines. Kampke (1986) discusses some special cases.

If SEPT and Smith's Rule are no longer optimal on parallel machines, two questions arise: What is the optimal policy and how far are SEPT and Smith's Rule from it.

On the first question, I believe the search for the optimal policy in the general case is futile. First, the problem is NP-hard; it seems likely that one can find a parametric family of distribution for which even minimization of expected flowtime is NP-hard. But the difficulties go beyond the computational effort involved: if SEPT or Smith's Rule are not optimal then the optimal policy may be extremely complicated to describe and to implement - it may involve dynamic scheduling with inserted idle time, and may depend on the entire processing time distribution of each job rather than on a few parameters. Needless to say, the data to estimate these will rarely be available in practice. Pinedo and Weiss (1987) discuss some very simple problems which have quite complicated optimal solutions; it is easy to imagine problems with optimal solutions of almost any degree of complexity.

How far are SEPT and Smith's rule from optimal can be measured in two ways: in terms of the expected value of the objective function, or in terms of the differences between the policies. In the present paper we give a complete answer in terms of the objective function. This answer is extremely favorable to Smith's Rule and SEPT.

Using the weight, the mean processing time, and one additional simply calculated parameter of the processing time distribution of each of the jobs, we calculate a bound on the difference between the expected objective function values. As long as the values of these three relevant parameters remain bounded, the bound on the expected difference does not grow with the number of jobs n. Thus under the assumption that the weights and the processing time distributions of all the jobs satisfy some uniform boundedness conditions we have that even though the expected (weighted) flowtime increases as $O(n^2)$, the expected difference in objective value between SEPT (or Smith's Rule) and the optimal policy is bounded by a constant, independent of n.

The uniform boundedness is clearly essential here, without it the worst case performance ratio for deterministic jobs is ~1.2, so the difference is $O(n^2)$. However, an assumption of uniform boundedness on processing time distributions is in general much less restrictive than for the deterministic case. In many problems all that will be required is uniform bounds on the means and variances of the processing times: this still permits any mixture of actual processing time values to occur and seems a reasonable requirement.

The results of this paper are derived in sections 4-11. While deriving the bounds for the objective function we obtain some insights into the nature of parallel processing, and some useful formulas for expected flowtime and weighted flowtime. The results are summarized in section 2, and a plausibility argument is given in section 3.

With the question of how well SEPT or Smith's Rule will do in terms of the expected value of the objective function settled, it remains to ask how different are the actual policies from the optimal. The plausibility argument indicates the SEPT may not be optimal towards the end of the schedule. We conjecture that SEPT and Smith's Rule have some turnpike property - asymptotically, for large n, most of the optimal decisions will be according to SEPT (or Smith's Rule). Such a turnpike result does indeed hold, and will be the subject of a forthcoming paper. A turnpike optimality result for a special case of a preemptive scheduling problem has been proven in Coffman, Hofri and Weiss (1988).

We conclude this paper with a discussion of some possible extensions to preemptive scheduling problems, the relation to Gittins index and to various control problems in section 12.

2. Summary of Results

Jobs $1,\dots,n$ require processing times $X_1,\dots,X_n$, nonnegative, which are drawn from some given probability distribution functions $F_1,\dots,F_n$ and are not known in advance. We assume nothing special about the form of $F_1,\dots,F_n$; We let $\mu_1,\dots,\mu_n$ and $\sigma_1^2,\dots,\sigma_n^2$ be their means and variances, and assume finite third moments. Machines $0,\dots,M$

become available to start the processing of these jobs at times $U_{00},\dots,U_{M0}$, with $U_{00}+\dots+U_{M0}=0$ and $S_0^2=\Sigma_0^M U_{i0}^2 / M$, and jobs are then processed by the machines in parallel, with no preemptions and no inserted idle times. Let $C_1,\dots,C_n$ be the completion times of the jobs. Let $U_{0n},\dots,U_{Mn}$ be the times at which the machines finish their processing; let $S_n^2=\{\Sigma_0^M U_{in}^2-\frac{1}{M+1}(\Sigma_0^M U_{in})^2\}/M$. Two objective functions are of interest: the flowtime $\sum_{j=1}^{n} C_j$ which is the sum of all the waiting times and all the processing times of all the jobs, and, more generally, the weighted flowtime, $\sum_{j=1}^{n} W_jC_j$ where the jobs are weighted by individual costs per unit time, $W_1,\dots,W_n$. Two policies suggest themselves for these objective functions: SEPT - start jobs in the order of shortest expected processing time first, and the so called "Smith's Rule" denoted by SR - start jobs in decreasing order of weight to expected processing time ratio. On a single processor, SEPT and SR minimize the expected flowtime and the expected weighted flowtime respectively. On M+1 parallel processors this is not generally the case.

In the present paper we prove that while SEPT and SR may not be optimal, they are very nearly optimal. Proceeding from the special to the general we obtain the following results:

(a) If all the jobs are identically distributed, the expected flowtime is:

$$E\left(\sum_{j=1}^{n} C_j\right)=\frac{n(n+1)}{2(M+1)}\mu+\frac{n\,M}{2(M+1)}\mu\left(1-\frac{\sigma^2}{\mu^2}\right)$$
$$-\frac{M}{2}\left(\frac{S_0^2}{\mu}-\frac{M}{M+1}\frac{E(X^3)}{3\mu^2}+\frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right)$$
$$+\frac{M}{2}\left(\frac{E(S_n^2)}{\mu}-\frac{M}{M+1}\frac{E(X^3)}{3\mu^2}+\frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\right) \qquad (2.1)$$

where the first $O(n^2)$ term is the single machine flowtime speeded up (M+1) times, the second O(n) term incorporates steady state delays per job caused by parallel processing, the third O(1) term corresponds to non steady state initial conditions, and the last term, which is o(1) and goes to zero as $n \to \infty$, corresponds to non steady state final conditions.

(b) If all the jobs have the same mean, the expected flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{2(M+1)} \mu + \frac{n\,M}{2(M+1)} \mu \left(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\,\mu^2}\right) \tag{2.2}$$

$$- \frac{M}{2} \frac{S_o^2}{\mu} + \frac{M}{2} \frac{E(S_n^2)}{\mu} .$$

In this expression, the only part which depends on the schedule is $E(S_n^2)$. Apart from that term and the initial conditions given by S_0^2, the expression depends only on μ and on $\sum_{j=1}^{n} \sigma_j^2/n$.

(c) For generally distributed jobs, we can obtain the expected value of the weighted flowtime, if the weights are equal to the expected processing times:

$$E(\sum_{j=1}^{n} \mu_j C_j) = \frac{1}{2(M+1)} \left(\sum_{j=1}^{n} \mu_j\right)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 \tag{2.3}$$

$$- \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2).$$

Again, the only term which depends on the schedule is $E(S_n^2)$.

(d) Consider now several classes of strategies and let:

Π_0 - the strategy of starting jobs in the arbitrary given order 1,..., n with no inserted idle time.

Π_1 - optimal list scheduling strategy.

Π_2 - optimal dynamic scheduling strategy.

Π_3 - optimal dynamic scheduling strategy with inserted idle time.

Π_4 - optimal dynamic scheduling strategy when the actual value of the processing time of a job becomes known when its processing starts.

Then,

$$E(\Sigma W_jC_j|\Pi_0) \geq E(\Sigma W_jC_j|\Pi_1) \geq E(\Sigma W_jC_j)|\Pi_2) \qquad (2.4)$$
$$\geq E(\Sigma W_jC_j|\Pi_3) \geq E(\Sigma W_jC_j|\Pi_4)$$

(e) If we take the special weights $\mu_1,\ldots,\mu_n$, (2.3) holds for Π_0 as well as Π_4 and so:

$$0 \leq E(\sum_{j=1}^{n} \mu_jC_j|\Pi_0) - E(\sum_{j=1}^{n} \mu_jC_j|\Pi_4) \leq \frac{M}{2} E(S_n^2|\Pi_0). \qquad (2.5)$$

Applying (2.3) and (2.4) for the subset of jobs $1,\ldots,k$, $1 \leq k \leq n$ we get for an arbitrary strategy Π from any of the classes of strategies in d:

$$E(\sum_{j=1}^{k} \mu_jC_j|\Pi_0) - E(\sum_{j=1}^{k} \mu_jC_j|\Pi) \leq \frac{M}{2} E(S_k^2|\Pi_0). \qquad (2.6)$$

Here S_k^2 is defined similar to S_n^2, as the sample variance of the times at which machines finish jobs $1,\ldots, k$ under Π_0.

(f) Let

$$\bar{D}^2 = \max_{1\leq j\leq n} \max_{s>0} \int_s^\infty (x-s)^2 \, dF_j(x). \qquad (2.7)$$

Then, if $S_0^2 = 0$,

$$E(S_k^2) \leq \frac{M}{M+1} \bar{D}^2 \qquad k = 1,\ldots,n \qquad (2.8)$$

(g) Let $\frac{W_1}{\mu_1} \geq \frac{W_2}{\mu_2} \dots \geq \frac{W_n}{\mu_n}$, then we can rewrite:

$$\sum_{j=1}^{n} W_j C_j = \frac{W_n}{\mu_n} \sum_{j=1}^{n} \mu_j C_j + \left(\frac{W_{n-1}}{\mu_{n-1}} - \frac{W_n}{\mu_n}\right) \sum_{j=1}^{n-1} \mu_j C_j \tag{2.9}$$

$$+ \cdot \cdot \cdot \cdot + \left(\frac{W_1}{\mu_1} - \frac{W_2}{\mu_2}\right) \sum_{j=1}^{1} \mu_j C_j .$$

(h) Substituting (2.3) in (2.9) we get:

$$E\left(\sum_{j=1}^{n} W_j C_j\right) = \frac{1}{M+1} E\left(\sum_{j=1}^{n} W_j C_j \,\middle|\, \text{one machine}\right)$$

$$+ \frac{M}{2(M+1)} \sum_{j=1}^{n} W_j \mu_j \left(1 - \frac{\sigma_j^2}{\mu_j^2}\right) \tag{2.10}$$

$$+ \frac{M}{2} \sum_{j=1}^{n} \frac{W_j}{\mu_j} [E(S_j^2) - E(S_{j-1}^2)]$$

Combining (2.6), (2.8) and (2.9) we then have:

Theorem:

$$E(\Sigma W_j C_j | SR) - E(\Sigma W_j C_j | \Pi_4) \leq \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D}^2. \tag{2.11}$$

3. A Plausibility Argument

The result of this paper is to show that the difference in the expected objective function when using SEPT or SR as a heuristic as against the optimal policy is, under some reasonable uniform boundedness conditions, bounded by a constant. In terms of the performance ratio

$$R = \frac{E(\text{objective}|\text{heuristic})}{E(\text{objective}|\text{optimal})}$$

this means that $R = 1+O(\frac{1}{n^2})$. In this section we discuss the plausibility of such a result.

As a first step it is easy to see that $R = 1+O(\frac{1}{\sqrt{n}})$. We use the following argument: Assume jobs are scheduled on M+1 processors in a particular given starting order. Consider a single processor which works at M+1 times the speed, and schedule the jobs in the same order. Then, for any realization, the completion times of each of the jobs on the M+1 processors and on the single processor differ by no more than M/(M+1) times the length of a single job. We therefore have:

<u>Proposition</u>: Let $E(X_1) \leq \ldots \leq E(X_n)$; Then

$$R = \frac{E(\text{Flowtime}|\text{SEPT})}{E(\text{Flowtime}|\text{OPT})} \leq 1+ \frac{nE(\max (X_1, \ldots, X_n))}{E(nX_1+ (n-1)X_2+\ldots + X_n)}.$$

If $\sum_{j=1}^{n} (n-j+1)\, EX_j \sim O(n^2)$, and $E(\max(X_1, \ldots, X_n)) \sim O(\sqrt{n})$ then $R = 1+ O(\frac{1}{\sqrt{n}})$.

A similar statement can be made for weighted flowtime and SR. We do not pursue this here since stronger results are derived in later sections of the paper. It is clear then that SEPT and SR are asymptotically optimal under uniform boundedness conditions; however, we claim much faster asymptotics than the above proposition indicates. The plausibility argument in this section tries to explain that.

Consider scheduling deterministic jobs to minimize flowtime. The flowtime on a single machine, when jobs are scheduled in the order $1, \ldots, n$ is $nX_1+(n-1)X_2+\ldots+ X_n$. In other words, during the processing of the 1st job, job 1, n jobs are waiting; during the processing of job j, (n-j+1) jobs are waiting, and during the processing of the last job only one job is waiting. Clearly we need to have $X_1 \leq X_2 \ldots \leq X_n$ for the optimal schedule.

For parallel machines, almost the same argument works, though it needs to be taken from last to first (see Figure 3.1)

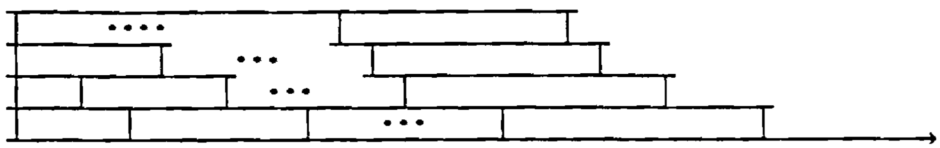


Figure 3.1: Flowtime on Parallel Machines.

Consider some schedule on M+1 parallel machines. Let J_1 be the set of all the jobs which are last on their machines; J_2 all the jobs which are before last, etc.: J_k are jobs which have k-1 jobs following them on the same machine. Assuming no idle times on any machine, the flowtime is: $\sum_{k=1}^{n} \sum_{j \in J_k} kX_j$. We therefore want to have the M+1 longest jobs in J_1, the next M+1 longest in J_2, etc; this yields the SPT schedule - start jobs according to shortest processing time first.

This argument does not work for stochastic processing times since the sets $J_1, J_2, \ldots$ cannot be assigned in advance.

For the sake of completeness we mention an alternative approach for the deterministic problem: Let $\tau_k < \tau_\ell$ be instants at which machines k and ℓ become available to start processing jobs; assume jobs are scheduled according to SPT; let n_k, n_ℓ be the number of jobs scheduled on machines k, ℓ respectively. Replace τ_k by $\tau_k+\Delta$ and τ_ℓ by $\tau_\ell-\Delta$; this will increase the flowtime by $\Delta(n_k-n_\ell)$. But $\Delta(n_k-n_\ell) \geq 0$ since under SPT, $n_k \geq n_\ell$. An inductive proof for optimality of SPT follows easily from this observation. This proof method can be adapted to stochastic processing times and is used by Weber, Varaiya and Walrand (1986) to show that SEPT is optimal if processing times are stochastically comparable.

Recall that X_i is stochastically greater than X_j, $X_i \geq_{ST} X_j$, if $F_i \leq F_j$ or equivalently, $Eh(X_i) \geq Eh(X_j)$ for every monotone increasing function h.

We return to the plausibility argument: The argument in favour of SEPT (as well as for Smith's Rule), is that at the beginning of the schedule there is a large number of jobs waiting and SEPT (or Smith's Rule) tend to reduce the number of jobs (their cost rate) fastest. This argument suffices to prove optimality for a single processor, and it applies to parallel processors as well. For parallel processors there exists, however, a counter argument: Towards the end of the schedule, as jobs are completed, there are no more new jobs to start and the processors fall idle one after the other; this means that processing at the end becomes inefficient and this of course has an effect on the objective function. Thus it seems that one ought to try to reduce these inefficiency periods. Minimization of these periods is hard (in the deterministic case, for two machines, it is equivalent to minimizing the makespan which is NP-hard), and it is not achieved by SEPT. If anything, it is asymptotically best to use LEPT to minimize makespan and the inefficiency periods (see Frenk and Rinnooy Kan, 1987). Nevertheless this inefficiency at the end is a boundary effect and is of marginal value - in particular it does not grow with the number of jobs n, so long as jobs remain uniformly bounded in some sense. Our conjecture is that this end effect is the only counter indication against optimality of SEPT (or Smith's Rule). Making this precise in some sense is the idea behind the proof in this paper.

4. Decomposition of Flowtime and of Weighted Flowtime

In this section we derive some useful formulas for decomposition of the flowtime and the weighted flowtime. Throughout this section we consider jobs 1,...,n which require processing times $X_1,\ldots,X_n$ and are started in the order 1,...,n; jobs are processed without preemptions and with no inserted idle times.

We let $C_1,\ldots,C_n$ denote their completion times. The results are for deterministic processing times and are of a combinatorial nature. We later regard them as sample path identities for the stochastic case. Much of the insight in these results follows by deriving them in easy steps: first for 1, then for 2, and finally for M+1 machines.

Case 1: Single Machine.

The completion time of job m is $C_m = X_1+\ldots+X_m$. Adding up we get for flowtime and weighted flowtime:

$$\sum_{j=1}^{n} C_j = nX_1+(n-1)X_2+\ldots+X_n \tag{4.1}$$

$$\sum_{j=1}^{n} W_jC_j = \sum_{j=1}^{n} \left(\sum_{k=j}^{n} W_k\right) X_j \tag{4.2}$$

Case 2: Two Machines:

We shall assume processing on the two machines starts at $U_0 \le V_0$ with $U_0+V_0 = 0$, and let $D_0 = V_0-U_0$. Let $U_m \le V_m$ be the times at which the two machines complete all the jobs 1,...,m, and let $D_m = V_m-U_m$. Clearly, $U_m+V_m = X_1+\ldots+X_m$. Job m+1 starts at U_m, and so:

$$C_{m+1} = U_m+X_{m+1} = (U_m+V_m)/2 - D_m/2 + X_{m+1} = \tag{4.3}$$

$$= \frac{1}{2}(X_1+\ldots+X_{m+1}) + \frac{1}{2}(X_{m+1}-D_m).$$

Adding up we get:

$$\sum_{j=1}^{n} C_j = \frac{1}{2} (n X_1+(n-1) X_2 +\ldots+ X_n) \tag{4.4}$$

$$+ \frac{1}{2} ((X_1-D_0) + (X_2-D_1) +\ldots+ (X_n-D_{n-1})).$$

and

$$\sum_{j=1}^{n} W_jC_j = \frac{1}{2} \sum_{j=1}^{n} (\sum_{k=j}^{n} W_k) X_j + \frac{1}{2} \sum_{j=1}^{n} W_j(X_j-D_{j-1}) \tag{4.5}$$

We recognize the first part in these formulas as the flowtime (weighted flowtime) for a single machine working at twice the speed of (4.1), (4.2). The second part consists of individual job delays caused by parallel processing.

We now consider the sequence of D_m's. At time U_m when job m+1 starts on one machine, D_m is the remaining processing time of the last of the jobs 1,...,m, which is running on the other machine. It is easy to see that

$$D_{m+1} = |X_{m+1}-D_m|. \tag{4.6}$$

If $X_1,\ldots, X_n$ are independent random variables, $\{D_m\}$ form a Markov chain.

The awkwardness of absolute value in (4.6) disappears when we square and sum over m.

$$\sum_{j=1}^{n} D_j^2 = \sum_{j=1}^{n} (X_j-D_{j-1})^2$$

from which we obtain the key formula

$$2 \sum_{j=1}^{n} X_jD_{j-1} = \sum_{j=1}^{n} X_j^2 + D_0^2 - D_n^2 \tag{4.7}$$

<u>Case 3</u>: <u>M+1 machines</u>

Let $U_{00} \leq U_{10} \leq \ldots \leq U_{M0}$ be the times at which the M+1 machines start processing, with $\sum_{i=0}^{M} U_{i0} = 0$, and, for $m = 1,\dots,n$, let $U_{0m} \leq U_{1m} \leq \dots \leq U_{Mm}$ be the times at which the M+1 machines complete the processing of jobs $1,\dots,m$; let $D_{im} = U_{im} - U_{0m}$, $i = 1,\dots,M$, $m = 0,\dots,n$.

Proposition 4.1: The flowtime on M+1 machines is

$$\sum_{j=1}^{n} C_j = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\, X_j + \frac{1}{M+1} \sum_{j=1}^{n} (MX_j - \sum_{i=1}^{M} D_{i\,j-1}) \tag{4.8}$$

and the weighted flowtime is:

$$\sum_{j=1}^{n} W_j C_j = \frac{1}{M+1} \sum_{j=1}^{n} (\sum_{k=j}^{n} W_k) X_j + \frac{1}{M+1} \sum_{j=1}^{n} W_j (MX_j - \sum_{i=1}^{M} D_{i\,j-1}) \tag{4.9}$$

Proof:

Job m+1 starts at U_{0m} and completes at $U_{0m} + X_{m+1}$. We have

$$\sum_{j=1}^{m} X_j = \sum_{i=0}^{M} U_{im} = (M+1) U_{0m} + \sum_{i=1}^{M} D_{im}.$$

Hence

$$C_{m+1} = \frac{1}{M+1} \sum_{j=1}^{m+1} X_j + \frac{M}{M+1} X_{m+1} - \frac{1}{M+1} \sum_{i=1}^{M} D_{im} \tag{4.10}$$

and (4.8), (4.9) follow. □

The Markovian recursion for D_{im}, $i=1,\dots,M$ is:

Proposition 4.2: The values $D_{1m+1} \leq \dots \leq D_{Mm+1}$ consist of the ordered values of the M largest among $X_{m+1}, D_{1m}, \dots, D_{Mm}$ minus the smallest among $X_{m+1}, D_{1m}, \dots, D_{Mm}$.

Proof: Follows from the fact that the completion time of jobs $1,\dots, m+1$ on the M+1 machines occur at $U_{0m} + X_{m+1}, U_{1m}, \dots, U_{Mm}$. □

The job processing time remainders D_{im}, $i = 1,\dots,M$, $m = 0,\dots,n$ can also be counted in a different way: Consider all the pairs of machines, $0 \le k < \ell \le M$; for a particular pair k,ℓ let $n'(k,\ell)$ be the number of jobs performed on these machines. Let $j' = 1,\dots,n'(k,\ell)$ be an index counting the jobs in their starting order, let $X_{j'}^{(k,\ell)}$ be the processing times, and let $D_{j'}^{(k,\ell)}$ $j' = 0,\dots,n'(k,\ell)$ be the remaining processing times all relative to the pair of machines (k,ℓ). The set of $D_{j'}^{(k,\ell)}$ $j' = 0,1,\dots,n'(k,\ell)$, $0 \le k < \ell \le M$ is exactly the set D_{im} $i = 1,\dots,M$, $m = 0,\dots,n$. To see this, consider $m = 0,\dots,n-1$, and look at the start of job $m+1$; assume it starts on machine k. On the other machines job remainders $D_{1m} \le \dots \le D_{Mm}$ are running; assume D_{im} is running on machine ℓ. Consider the pair of machines k,ℓ, let j' be the index of job $m+1$, relative to (k,ℓ) (in other words, job $m+1$ is the j' job that is started on the pair of machines k,ℓ); then D_{im} is $D_{j'-1}^{(k,\ell)}$. In particular:

$$\sum_{j=0}^{n} \sum_{i=1}^{M} D_{ij} = \sum_{0 \le k < \ell \le M} \sum_{j'=1}^{n'(k,\ell)} D_{j'}^{(k,\ell)} \tag{4.11}$$

<u>Proposition 4.3 - Key Formula</u>: Let $S_j^2 = \frac{1}{M} \sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)} \left(\sum_{i=1}^{M} D_{ij} \right)^2$ (this is the sample variance of $0, D_{1j}, \dots, D_{Mj}$ or equivalently $U_{0j}, \dots, U_{Mj}$), $j = 0, 1, \dots, n$. Then:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1) S_0^2 - M(M+1) S_n^2 \tag{4.12}$$

<u>Proof:</u> By the above argument:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1} = 2 \sum_{0 \le k < \ell \le M} \sum_{j'=1}^{n'(k,\ell)} X_{j'}^{(k,\ell)} D_{j'-1}^{(k,\ell)} . \tag{4.13}$$

Applying (4.7) to each pair of machines this equals

$$\sum_{0\le k<\ell\le M}\sum \left\{ \sum_{j'=1}^{n'(k,\ell)} (X_{j'}^{(k,\ell)})^2 + (D_0^{(k,\ell)})^2 - (D_{n'(k,\ell)}^{(k,\ell)})^2 \right\} \tag{4.14}$$

The summation over all pairs (k,ℓ) includes each X_j^2, $j = 1,\dots, n$, exactly M times. Also, $D_0^{(k,\ell)}$ equals $U_{k'o}-U_{\ell'o}$ for some pair of indices (k',ℓ') so that going over $0 \le k < \ell \le M$, $\{k',\ell'\}$ goes over all the pairs; Similarly, $D_{n'(k,\ell)}^{(k,\ell)}$ equals $U_{k''n}-U_{\ell''n}$. Hence (4.14) equals

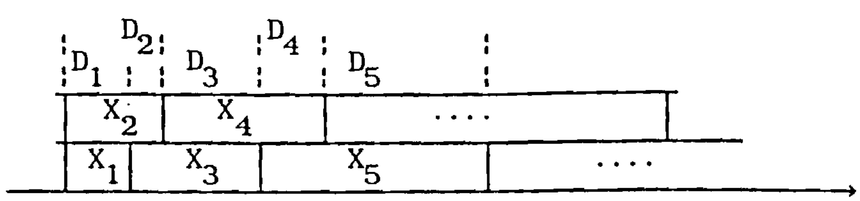
$$M \sum_{j=1}^{n} X_j^2 + \sum_{0\le k<\ell\le M}\sum \{(U_{k0}-U_{\ell 0})^2 - (U_{kn}-U_{\ell n})^2\} \tag{4.15}$$

Expression (4.12) follows from the well known statistics formula:

$$\sum_{1\le s<t\le L} (a_s-a_t)^2 = L \sum_{s=1}^{L} a_s^2 - (\sum_{s=1}^{L} a_s)^2 \tag{4.15}$$

<u>An Aside on SPT</u>:

The decompositions (4.4), (4.8) allow us to get explicit formulas for flowtime when jobs are scheduled according to SPT. We look first at two machines. We let $U_0 = V_0$, and for SPT we have $X_1 \le X_2 \le \dots \le X_n$. Then the jobs start on the two machines in alternating order, $D_1 \le D_3 \le \dots$, and $D_2 \le D_4 \le \dots$, and the successive periods of length D_j are consecutive and not overlapping, see Figure 4.1.



<u>Figure 4.1: SPT on two machines</u>

More precisely, with $D_0 = 0$, $D_1 = X_1$, and by induction $D_{j+1} = X_{j+1} - D_j$ (since if we assume $D_j = X_j - D_{j-1}$ then $D_j \le X_j \le X_{j+1}$); also $U_{j+1} = V_j$ $j = 0,\dots,n-1$. Hence $\sum_{j=0}^{n-1} D_j = U_n$, and $\sum_{j=1}^{n} (X_j - D_{j-1}) = V_n$. V_n is however the makespan and so, by (4.4) we have, for 2 machines.

$$(\text{Flowtime}|\text{SPT on 2 machines}) = \tag{4.17}$$
$$= \frac{1}{2}(\text{Flowtime}|\text{SPT on 1 machine}) + \frac{1}{2}(\text{Makespan}|\text{SPT on 2 machines}).$$

On M+1 machines we have:

Proposition 4.4: Let $U_{00} = \dots = U_{M0}$ be the starting times of the M+1 machines, let jobs be scheduled by SPT so that $X_1 \le \dots \le X_n$, and let $U_{0n} \le \dots \le U_{Mn}$ be the finishing times for the M+1 machines. Then

$$\sum_{j=1}^{n} C_j = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\, X_j + \frac{1}{M+1} \sum_{i=1}^{M} i\, U_{in} \tag{4.18}$$

Proof: We use (4.8) and (4.11). To obtain $\sum_j \sum_i (X_j - D_{i\,j-1})$ as in (4.8), we look at all pairs of machines (k,ℓ). For a pair (k,ℓ) we calculate $\sum (X_{j'}^{(k,\ell)} - D_{j'-1}^{(k,\ell)})$. The $n'(k,\ell)$ jobs on these two machines are again scheduled by SPT, so the sum equals the makespan $V_{n'(k,\ell)}^{(k,\ell)}$ which is the finishing time of the later of these two machines. Summing over all pairs of machines, U_{Mn} gets counted M times, and in general U_{in} gets counted i times. The proposition follows. □

5. Flowtime for Random IID Jobs.

In this section we consider jobs whose processing times $X_1,\dots,X_n$ are drawn independently from a common distribution F. For this case we obtain explicit formulas for the expected flowtime. We then apply these to get the expected flowtime for four specific examples of processing time distributions.

We let $\mu = E(X)$, $\sigma^2 = V(X)$ be the mean and variance of F, and assume $E(X^3) < \infty$; for simplicity we also assume in this section that F is non-arithmetic.

Imagine that there is an unlimited number of jobs which are scheduled on several parallel machines. The completion times of the jobs then form independent renewal processes. Consider an instant at which a job completion occurs on one machine, and examine the remaining processing times on the other machines. To be specific, let U_{i0}, $i = 0,\ldots,M$ be the initial starting times of the machines, and assume that a job is completed on machine k at time u. Observing the other machines at time u (conditional on $U_{10},\ldots, U_{M0}$ and on u), the remaining processing times on the other M machines, $\{D_{1m},\ldots,D_{Mm}\}$, are independent forward recurrence times of the renewal processes, and for $u \to \infty$ they converge to M independent random variables identically distributed with the equilibrium distribution. Let D_∞ denote a random variable from the equilibrium distribution; it has probability density function $f_e(x) = \bar{F}(x)/\mu$ (where $\bar{F}(x) = 1-F(x)$), and 1st and 2nd moments $(\mu^2+\sigma^2)/2\mu$ and $E(x^3)/3\mu$ (Cox 1962).

Some care is needed in the preceding description: The independence hinges on the fact that we condition only on u. Independence still holds if we take the random instance U at which the completion of the jth job on machine k occurs, and condition on j. However, given that the m'th job completion (on all processors) occurs at time u (fixed or random) on machine k, the remaining processing times on the other M machines are no longer independent; even conditioning only on $m \leq n$, where n is the total number of jobs, destroys the independence - this point was raised by Van der Wal and Hordijk, (1987). Nevertheless, as $m \to \infty$, the joint distribution of the remaining processing times on the other processors, $D_{1m} \leq \dots \leq D_{Mm}$, converges to that of an ordered sample from the equilibrium distribution. To prove this convergence we need to examine the Markov chain of $D_{1m}, \dots, D_{Mm}$, $m = 0,1,\dots$, for iid X_m. Feller (1971, Chapter VI, Section II example f, p. 208) discusses the chain $D_{m+1} = |X_{m+1}-D_m|$, and Cox and Miller (1965, pp. 362-365) briefly discuss the multivariate chain; we were unable though to find a complete statement of the following theorem in the literature.

<u>Theorem 5.1</u>: For F nonarithmetic $D_{1m},\dots,D_{Mm}$ converge in distribution as $m \to \infty$ to $D_{1\infty},\dots,D_{M\infty}$ with joint probability density function $f_e(\underline{x}) = \frac{M!}{\mu^M} \bar{F}(x_1)\dots \bar{F}(x_M)$, $x_1 \leq \dots \leq x_M$.

<u>Proof:</u> The proof is based on the approach in Feller, 1971, Chapter VIII, Section 7, pp. 270-274, which discusses ergodicity of Markov chains in discrete time with a general state space, and with a stationary transition kernel K(x,dy). Feller proves the following Theorem (Feller, ditto, theorem 2): A strictly positive regular kernel K is ergodic if and only if it possesses a strictly positive stationary probability distribution α. We use an adaptation of this theorem to show that the chain $D_{1m} \leq \dots \leq D_{Mm}$ is ergodic with convergence to the stated distribution $f_e(\underline{x})$. The proof requires the following steps:

(i) It is easily checked that $f_e(\underline{x})$ is the density of a strictly positive stationary distribution for the kernel K.

(ii) The kernel is regular (as defined in Feller, ditto, definition 3) - this follows easily from the fact that our transition kernel operates like a convolution.

(iii) While for general F nonarithmetic the kernel is not strictly positive (as defined in Feller, ditto, definition 1) it can be shown that it is asymptotically strictly positive in the sense:

Definition: The kernel $K(\underline{x}, d\underline{y})$ is asymptotically strictly positive in $\Omega \subseteq R^M$ if for every $\epsilon > 0$ there exists N so that for all $n > N$, $K^{(n)}(\underline{x}, I) > 0$ whenever the point $\underline{x}$ and the interval I satify: $\underline{x} \in \Omega \cap (0, 1/\epsilon)^M$, $I \subseteq \Omega \cap (0, 1/\epsilon)^M$, and $|P_i(I)| > \epsilon$ $i = 1, \ldots, M$ ($P_i(I)$ is the projection of I on the i'th coordinate). This fact follows again from the convolution like nature of the kernel, utilizing a proof similar to Feller, Chapter V, section 4a, pp. 147-148.

(iv) The proof of Feller's theorem 2 can be modified to work when strictly positive is replaced by asymptotic strictly positive. The proof works in R^M as well as in R.

The proof is rather technical and given in the Appendix. □

We now turn to the value of $E(\sum_{j=1}^{n} \sum_{i=1}^{M} D_{ij-1})$ which appears in the flowtime expression (4.8). By Theorem 5.1 $E(\sum_{i=1}^{M} D_{im}) \to M(\sigma^2+\mu^2)/2\mu$ as $m \to \infty$. In fact the convergence is fast enough so that the deviations from the limit form a convergent series. We have:

Proposition 5.2:

$$\lim_{n\to\infty} E \sum_{j=1}^{n} \sum_{i=1}^{M} \left(D_{ij-1} - \frac{\sigma^2+\mu^2}{2\mu}\right) =$$

$$= \frac{M(M+1)}{2} \frac{S_0^2}{\mu} - \frac{M^2 E(X^3)}{6\mu^2} + \frac{M(M-1)(\sigma^2+\mu^2)^2}{8\mu^3} \tag{5.1}$$

Proof: We use the key formula (4.12), and take expectation on both sides:

$$E\,2\sum_{j=1}^{n}\sum_{i=1}^{M} X_j D_{ij-1} = E\{\,M\sum_{j=1}^{n} X_j^2 + (M+1)S_0^2 - M(M+1)\,S_n^2\,\}. \qquad (5.2)$$

On the left hand side we note that D_{ij-1} are functions of $X_1,\ldots,X_{j-1}$ only and so (whether j is fixed in advance or dependent on D_{ij-1}), X_j is distributed like F, independent of D_{ij-1}, i=1,..., M, with $E\,X_j = \mu$. Hence:

$$2\mu\,E\sum_{j=1}^{n}\sum_{i=1}^{M} D_{ij-1} = n\,M(\sigma^2+\mu^2) + M(M+1)S_0^2 - M(M+1)E\,S_n^2. \qquad (5.3)$$

It remains to obtain $\lim_{n\to\infty} E\,S_n^2$, which by theorem 5.1 is:

$$\lim_{n\to\infty} E\,S_n^2 = \frac{1}{M}E\sum_{i=1}^{M} D_{i\infty}^2 - \frac{1}{M(M+1)}E\,(\sum_{i=1}^{M} D_{i\infty})^2 \qquad (5.4)$$

$$= \frac{M}{M+1}E(D_\infty^2) - \frac{M-1}{M+1}(E\,D_\infty)^2 = \frac{M}{M+1}\frac{E(X^3)}{3\mu} - \frac{M-1}{M+1}(\frac{\sigma^2+\mu^2}{2\mu})^2,$$

and (5.1) follows. □

We combine the preceding results to obtain the main result of this section:

<u>Theorem 5.3</u>: The expected flowtime of n iid jobs on M+1 parallel machines is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{2(M+1)}\mu + \frac{n\,M}{2(M+1)}\mu\,(1-\frac{\sigma^2}{\mu^2})$$

$$-\frac{M}{2}\{\frac{S_0^2}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\} \qquad (5.5)$$

$$+\frac{M}{2}\{\frac{E(S_n^2)}{\mu} - \frac{M}{M+1}\frac{E(X^3)}{3\mu^2} + \frac{M-1}{M+1}\frac{(\mu^2+\sigma^2)^2}{4\mu^3}\}$$

<u>Proof</u>: We rewrite the decomposition formula (4.8) and take expectations as follows:

$$E\left(\sum_{j=1}^{n} C_j\right) = E\,\frac{1}{M+1}\sum_{j=1}^{n}(n-j+1)\,X_j$$

$$+\,E\,\frac{1}{M+1}\sum_{j=1}^{n}\left(M\,X_j - \sum_{i=1}^{M} D_{i\infty}\right)$$

$$-\,E\,\frac{1}{M+1}\sum_{j=1}^{\infty}\left(\sum_{i=1}^{M} D_{i\,j-1} - \sum_{i=1}^{M} D_{i\infty}\right) \tag{5.6}$$

$$+\,E\,\frac{1}{M+1}\sum_{j=n+1}^{\infty}\left(\sum_{i=1}^{M} D_{i\,j-1} - \sum_{i=1}^{M} D_{i\infty}\right).$$

Substituting the expected value of X_j, of D_∞, and formula (5.1) we obtain (5.5). □

Note that the first term in (5.5) is the single machine flowtime for a machine with M+1 fold speed. The second term contains an $\frac{M}{M+1}\mu\,(1 - \frac{\sigma^2}{\mu^2})$ delay per job that is the effect of parallel processing. If processing could start from stationary conditions this would be all. It is seen from (5.6) that the last two terms are the effect of starting in nonstationary conditions and of ending in nonstationary conditions. By (5.1) the last term converges to 0 as $n \to \infty$. The four terms are therefore $O(n^2)$, $O(n)$, $O(1)$ and $o(1)$ respectively. The whole expression is a function of μ, σ^2, S_0^2 and $E(S_n^2)$. Only the last of these depends on the form of the distribution. The following four examples further illustrate the theorem.

<u>Example 1 - Deterministic Jobs</u>: $X_j = 1$, so $\sigma^2/\mu^2 = 0$. Assume jobs start with intervals of $\frac{1}{M+1}$ between them at $\frac{-M}{2(M+1)}$, $\frac{-M}{2(M+1)} + \frac{1}{M+1}$, $\frac{-M}{2(M+1)} + \frac{2}{M+1}$, ..., $\frac{-M}{2(M+1)} + \frac{M}{M+1}$; note that the sum of starting times on the M+1 machines is 0. Job completion times are at $\frac{M}{2(M+1)} + \frac{1}{M+1}$, $\frac{M}{2(M+1)} + \frac{2}{M+1}$, $\frac{M}{2(M+1)} + \frac{3}{M+1}$,.... On a single M+1 fold speed machine completions would

be at $\frac{1}{M+1}, \frac{2}{M+1}, \frac{3}{M+1}, \ldots$. Thus each job is delayed by $\frac{M}{2(M+1)}$. This delay per job is the maximal possible for any distribution F. Note that here $D_{im} = \frac{i}{M+1}$ and $S_0^2 = S_n^2$. See Figure 5.1.

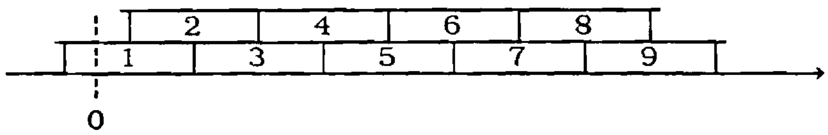


Figure 5.1: Deterministic Jobs on 2 Parallel Machines, n = 9.

The general formula is

$$E(\Sigma C_j) = \frac{n(n+M+1)}{2(M+1)}. \quad (5.7)$$

Example 2 - Exponential jobs: $X_j \sim \exp(1)$, $\sigma^2/\mu^2 = 1$. Assume jobs start at 0. While the machines are all busy successive job completions occur with ~ exp(M+1) intervals, exactly as on a M+1-fold speed single machine. There is no steady state delay per job - this is true whenever $\frac{\sigma^2}{\mu^2} = 1$. Here we chose $S_0^2 = 0$. D_{im}, i= 1,...M are an ordered sample from an exp(1) distribution, and $D_{im} \sim D_{i\infty}$. See Figure 5.2.

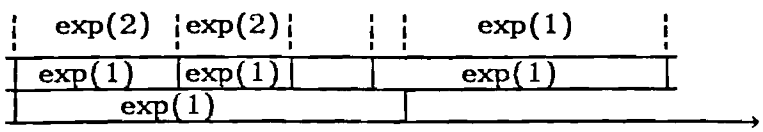


Figure 5.2: Exponential Jobs on 2 Parallel Machines, n = 4

The exact formula is:

$$E(\Sigma C_j) = \frac{n(n+1)}{2(M+1)} + \frac{M}{2}. \quad (5.8)$$

Example 3 - Uniform jobs: $X \sim U(0,2)$, $E(X) = 1$, $\sigma^2/\mu^2 = \frac{1}{3}$. For this distribution, D_∞ has p.d.f. $f_e(x) = 1 - \frac{1}{2}x$, $0 \leq x \leq 2$, and what's more, starting all the machines at time 0, the first job completion at a time > 0 already has D_{im} distributed like $D_{i\infty}$: see Feller, ditto, problem 22,

p. 217. The delay per job is $\frac{1}{3}\frac{M}{M+1}$. The formula for the flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n + \frac{2}{3}M + 1)}{2(M+1)} + \frac{M(M+2)}{9(M+1)} \tag{5.9}$$

Example 4 - DHR jobs: If X_j have a DHR (decreasing hazard rate, $h(x) = f(x)/\bar{F}(x)$ is decreasing), then $\sigma^2/\mu^2 > 1$. In that case the delay per job is negative. That is to say, the expected flowtime on M+1 parallel machines is smaller than on a single M+1 fold speed machine. This agrees with the result of Weber (1982) that in preemptive scheduling of DHR jobs processor sharing is optimal.

In general, the delay per job in comparison to a single M+1 fold faster machine is $\frac{M}{M+1}(1 - \frac{\sigma^2}{\mu^2})$. In the above examples we looked at $\frac{\sigma^2}{\mu^2} = 0, \frac{1}{3}, 1, >1$. The first three examples are special in that D_{im} actually become equal to $D_{i\infty}$, and so explicit formulas are obtained.

6. Jobs with Equal Mean Processing Times

In this section we consider jobs whose processing times $X_1,\dots,X_n$ are drawn independently from distributions $F_1,\dots,F_n$, all of them possessing the same mean $E(X_j) = \mu$; the jobs may however have different variances, $V(X_j) = \sigma_j^2$. On a single machine, the expected flowtime is $\frac{n(n+1)}{2}\mu$ for every nonpreemptive work conserving schedule. The minimization of expected flowtime on M+1 parallel machines is hard - they do not fall under the category of stochastically comparable jobs (as in Weber, Varaiya and Walrand 1986) and of course the SEPT rule is meaningless for such jobs, since every policy is SEPT. Pinedo and Weiss (1987) show that in some cases LVF (Largest Variance First) is optimal, but not in general.

We start by deriving a generalization of formula (5.5) for the expected flowtime. From this formula it becomes evident that the difference in expected flowtime between any two schedules lies in the "end effect" as conjectured in section 3, and the difference is bounded by an O(1) quantity. We then discuss a connection with the minimization of expected makespan, which emerges from the formulas.

Theorem 6.1: The expected flowtime for n jobs with mean processing time μ and variances σ_j^2, $j = 1,\dots,n$ scheduled on M+1 machines in the starting order $1,\dots, n$ is given by:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{2(M+1)}\mu + \frac{nM}{2(M+1)}\mu\Big(1 - \frac{\sum_{j=1}^{n}\sigma_j^2}{n\mu^2}\Big) - \frac{M}{2}\frac{S_0^2}{\mu} + \frac{M}{2}\frac{E(S_n^2)}{\mu}. \tag{6.1}$$

Proof: We use the key formula (4.12), and take expectations on both sides, as in (5.2):

$$E\, 2\sum_{j=1}^{n}\sum_{i=1}^{M} X_j D_{ij-1} = E\{M\sum_{j=1}^{n} X_j^2 + M(M+1)S_0^2 - M(M+1)S_n^2\}. \tag{6.2}$$

Conditional on j, X_j is independent of D_{ij-1} and for all j, $E(X_j) = \mu$. Hence, (with $EX_j^2 = \mu^2 + \sigma_j^2$):

$$E\sum_{j=1}^{n}\sum_{i=1}^{M} D_{ij-1} = \frac{M\sum_{j=1}^{n}(\mu^2+\sigma_j^2)}{2\mu} + \frac{M(M+1)}{2\mu}S_0^2 - \frac{M(M+1)}{2\mu}(S_n^2) \tag{6.3}$$

Taking expectations on the decomposition formula (4.8) we have:

$$E(\sum_{j=1}^{n} C_j) = E\{\frac{1}{M+1}\sum_{j=1}^{n}(n-j+1)\, X_j + \frac{M}{M+1}\sum_{j=1}^{n} X_j - \frac{1}{M+1}\sum_{j=1}^{n}\sum_{i=1}^{M} D_{ij-1}\}. \tag{6.4}$$

Substituting $EX_j = \mu$ and (6.3) we obtain (6.1). □

As in (5.5), the first term in (6.1) is the expected flowtime for a single M+1 fold speed machine, the second term contains per job delays, with $\Sigma\, \sigma_j^2/n$ replacing σ^2 for the average delay, the third term is the effect of initial conditions and the last term is the end effect of the idle times at the end of the schedule. Only the last term depends on the form of the distributions $F_1, \ldots, F_n$, and only the last term depends on the schedule.

Consider any two scheduling strategies which are nonpreemptive and work conserving (i.e. do not allow idle time while unstarted jobs are available), Π_0 and Π, then:

$$E\left(\sum_{j=1}^{n} C_j \,|\, \Pi_0\right) - E\left(\sum_{j=1}^{n} C_j \,|\, \Pi\right) \leq \frac{M}{2\mu} E(S_n^2 \,|\, \Pi_0). \tag{6.5}$$

Expression (6.5) indicates that the difference in expected flowtime between any two schedules is bounded. We return to this bound in section 10.

Since the transition probabilities of the Markov chain of $D_{1m}, \ldots, D_{Mm}$ are no longer stationary, the asymptotic behaviour of S_n is no longer as simple as in the i.i.d. case. Nevertheless, it is clear that as $n \to \infty$, the dependence of S_n on the first n_0 steps in the schedule tends to disappear.

From expression (6.1) it is seen that to minimize the expected flowtime is equivalent to minimizing $E(S_n^2)$.

Recall the definition of S_j^2 in terms of D_{ij} $i = 1, \ldots, M$; S_n^2 is the sample variance of $0, D_{1j}, \ldots, D_{Mj}$ and it is seen immediately that it can also be written as:

$$S_j^2 = \frac{1}{M} \sum_{i=0}^{M} U_{ij}^2 - \frac{1}{M(M+1)} \left(\sum_{i=0}^{M} U_{ij}\right)^2 \tag{6.6}$$

We can now express the expected flowtime in terms of the machine completion times $U_{0n} \leq U_{1n} \leq \ldots \leq U_{Mn}$:

<u>Proposition 6.2:</u> The expected flowtime can be written as:

$$E(\sum_{j=1}^{n} C_j) = \frac{1}{2\mu} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{2\mu} \sum_{i=0}^{M} U_{i0}^2 + \frac{n}{2}\mu(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\mu^2}) \qquad (6.7)$$

<u>Proof:</u> Since we assume $\sum_{i=0}^{M} U_{i0} = 0$, we have $S_0^2 = \frac{1}{M} \sum_{i=0}^{M} U_{i0}^2$. To obtain an alternative expression for $E(S_n^2)$ we note that

$$\sum_{i=0}^{M} U_{in} = \sum_{j=1}^{n} X_j \qquad (6.8)$$

so that

$$E(S_n^2) = \frac{1}{M} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{M(M+1)} E(\sum_{i=0}^{M} U_{in})^2 =$$

$$= \frac{1}{M} E(\sum_{i=o}^{M} U_{in}^2) - \frac{1}{M(M+1)} E(\sum_{j=1}^{n} X_j)^2 \qquad (6.9)$$

$$= \frac{1}{M} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{M(M+1)} (n^2\mu^2 + \sum_{j=1}^{n} \sigma_j^2).$$

Substituting in (6.1) we obtain (6.7). □

For two machines, M= 1, we have $S_n^2 = \frac{1}{2} D_n^2$ where D_n is the remaining processing time between the last two completion times. The expected flowtime is:

$$E(\sum_{j=1}^{n} C_j) = \frac{n(n+1)}{4}\mu + \frac{n}{4}\mu(1 - \frac{\sum_{j=1}^{n} \sigma_j^2}{n\mu^2}) - \frac{1}{4}\frac{D_0^2}{\mu} + \frac{1}{4}\frac{E(D_n^2)}{\mu}$$

on the other hand the makespan is:

$$C_{max} = \frac{1}{2} \sum_{j=1}^{n} X_j + \frac{1}{2} D_n$$

and so:

$$E(C_{max}) = \frac{n}{2}\mu + \frac{1}{2} E(D_n). \qquad (6.10)$$

Hence, minimization of expected flowtime is equivalent to minimizing $E(D_n^2)$ while minimization of expected makespan is equivalent to minimizing $E(D_n)$. For many special cases, the same policy minimizes both, e.g. all the examples in Pinedo and Weiss (1987). For M+1 machines, we obtain:

<u>Corollary 6.3:</u> Minimization of expected flowtime is equivalent to minimization of the expected squared L_2 norm of $U_{0n}, U_{1n}, \ldots, U_{Mn}$.

In other words, when all $E(X_j)=\mu$, the minimization of expected flowtime is equivalent to a version of a stochastic makespan minimization problem.

For the deterministic problem minimizing makespan is the same as minimizing D_{1n}, which is the same as minimizing $S_n^2 = \frac{1}{2} D_n^2$ or minimizing $U_{0n}^2 + U_{1n}^2$. For more than two machines several functions of $U_{0n}, U_{1n}, \ldots, U_{Mn}$, the machine completion times, can be taken as generalizations of the two machine makespan. The natural one is to take $C_{max} = \max(U_{0n}, \ldots, M_{Mn}) = U_{Mn}$, which is the L_∞ norm of $U_{0n}, \ldots, U_{Mn}$. However, the L_2 norm, $\{\sum_{i=0}^{M} U_{in}^2\}^{1/2}$ is also a sensible measure for machine utilization. In the stochastic case the L_∞ and L_2 norms are random variables and we can try and minimize their expectations; for L_2, as often happens, minimization of the expected squared norm is more tractable. Our result here is that for equal mean processing times it is equivalent to minimizing flowtime.

7. <u>A Special Case of Weighted Flowtime.</u>

In this section we consider jobs with general random processing times, $X_1 \ldots, X_n$ with distribution functions $F_1, \ldots, F_n$, means $\mu_1, \ldots, \mu_n$ and variances $\sigma_1^2, \ldots, \sigma_n^2$. The simplicity of formulas (5.5) and (6.1) does not carry over to this general case. However, if we consider weighted flowtime, with weights which are equal to the mean processing times, a straightforward generalization of (5.5), (6.1) is possible:

Theorem 7.1: The expected weighted flowtime when C_j - the completion time of job j, is weighted by μ_j - the expected processing time of job j is, for every nonpreemptive work conserving schedule:

$$E(\sum_{j=1}^{n} \mu_j C_j) = \frac{1}{2(M+1)} (\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 - \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2) \qquad (7.1)$$

Proof: Assume first that jobs are started in the predetermined order 1,...,n. Taking expectations in the key formula (4.12) we have:

$$2E(\sum_{j=1}^{n} \sum_{i=1}^{M} X_j D_{ij-1}) = M\, E(\sum_{j=1}^{n} X_j^2) + M(M+1)\, S_0^2 - M(M+1)\, E(S_n^2) \qquad (7.2)$$

Because the order is predetermined, and D_{ij-1} depends on $X_1, \ldots X_{j-1}$ only, X_j is independent of D_{ij-1}, $i = 1, \ldots, M$. Therefore:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} \mu_j\, E\, D_{ij-1} = M \sum_{j=1}^{n} (\mu_j^2+\sigma_j^2) + M(M+1)\, S_0^2 - M(M+1)\, E(S_n^2) \qquad (7.3)$$

Taking expectation in the decomposition formula (4.9)

$$E \sum_{j=1}^{n} \mu_j C_j = \frac{1}{M+1} E \{ \sum_{j=1}^{n} (\sum_{k=j}^{n} \mu_k)\, X_j\} + \frac{M}{M+1} \sum_{j=1}^{n} \mu_j E(X_j) - \frac{1}{M+1} \sum_{j=1}^{n} \sum_{i=1}^{M} \mu_j\, E(D_{ij-1}) \qquad (7.4)$$

The first expectation is equal to $\sum_{j=1}^{n} \sum_{k=j}^{n} \mu_k\, \mu_j = \frac{1}{2} (\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2$, and further substitution yields (7.1).

Consider now a more general scheduling rule, under which the jobs are started in the order J(1), J(2),..., J(n), some permutation of 1,..., n. Here J(1), J(2),... can be random, and the j'th job to start, J(j) can depend on J(1),... J(j-1) and on the values of $X_1, \ldots, X_{j-1}$ on which information is already available when the j'th job start occurs.

The decomposition formula (4.9) now reads:

$$\sum_{j=1}^{n} \mu_{J(j)} C_{J(j)} = \frac{1}{M+1} \sum_{j=1}^{n} \left(\sum_{k=j}^{n} \mu_{J(k)} \right) X_{J(j)} \tag{7.5}$$

$$+ \frac{1}{M+1} \sum_{j=1}^{n} \mu_{J(j)} \left(M X_{J(j)} - \sum_{i=1}^{M} D_{i\,j-1} \right)$$

The expectation of the first term adds up to:

$$\frac{1}{2(M+1)} \left(\sum_{j=1}^{n} \mu_j \right)^2 + \frac{1}{2(M+1)} \sum_{j=1}^{n} \mu_j^2 .$$

Also, by conditioning,

$$\begin{aligned} E(X_{J(j)} D_{i\,j-1}) &= E[D_{i\,j-1}\ E(X_{J(j)} | D_{i\,j-1})] \\ &= E[D_{i\,j-1}\ E\{\ E(X_{J(j)} | D_{i\,j-1}, J(j))\ | D_{i\,j-1}\} \qquad (7.6) \\ &= E[D_{i\,j-1}\ E(\mu_{J(j)} | D_{i\,j-1})] = E(\mu_{J(j)} D_{i\,j-1}). \end{aligned}$$

where the main point is that

$$E(X_{J(j)} | J(j),\ D_{i\,j-1}) = E(X_{J(j)} | J(j)) = \mu_{J(j)} . \tag{7.7}$$

The key formula (4.12) now reads:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} (X_{J(j)} D_{i\,j-1}) = M \sum_{j=1}^{n} X_{J(j)}^2 + M(M+1) S_o^2 - M(M+1)\ E(S_n^2),$$

and taking expectations we have, by (7.6),

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} E(\mu_{J(j)} D_{i\,j-1}) = \mu \sum_{j=1}^{n} (\mu_j^2 + \sigma_j^2) + M(M+1) S_o^2 - M(M+1) E(S_n^2). \tag{7.8}$$

Taking expectations in (7.5) and substituting (7.8) we again obtain (7.1). □

Several corollaries are easily obtained here:

Using the processing times X_j rather than their means μ_j as weights we have:

Corollary 7.2: For the case when C_j, the completion time of job j is weighted by X_j,

$$E(\sum_{j=1}^{n} X_j C_j) = E(\sum_{j=1}^{n} \mu_j C_j) + \sum_{j=1}^{n} \sigma_j^2 \tag{7.9}$$

Proof: Consider (7.5); clearly in the new expected objective function $E(\sum_{j=1}^{n} X^2_{J(j)})$ replaces $E(\sum_{j=1}^{n} \mu_{J(j)} X_{J(j)})$. at the same time, $E(X_{J(j)} X_{J(k)}) = E(\mu_{J(j)} \mu_{J(k)})$ for k > j, and $E(X_{J(j)} D_{ij-1}) = E(\mu_{J(j)} D_{ij-1})$, so nothing else is changed. (7.9) follows. □

With μ_j as weight for C_j, j = 1,...,n, the expected weighted flow-time on a single machine is independent of the schedule. For M+1 parallel machines we have in analogy with (6.5):

Proposition 7.3: For any two nonpreemptive work conserving strategies, Π_o, Π

$$E(\Sigma \mu_j C_j | \Pi_o) - E(\Sigma \mu_j C_j | \Pi) \le \frac{M}{2} E(S_n^2 | \Pi_o). \tag{7.10}$$

Proof: By (7.1). □

We discuss bounds for $E(S_n^2)$ in Section 10.

In analogy with proposition 6.2 we have:

Corollary 7.4: In terms of machine start times $U_{00} \le \dots \le U_{M0}$ and finish times $U_{0n} \le \dots \le U_{Mn}$:

$$E(\sum_{j=1}^{n} \mu_j C_j) = \frac{1}{2} E(\sum_{i=0}^{M} U_{in}^2) - \frac{1}{2} \sum_{i=0}^{M} U_{i0}^2 + \frac{1}{2}(\sum_{j=1}^{n} \mu_j)^2 - \frac{1}{2} \sum_{j=1}^{n} \sigma_j^2 \tag{7.11}$$

Proof: As in the proof of proposition 6.2, $S_0^2 = \frac{1}{M} \sum_{i=0}^{M} U_{i0}^2$ while

$$E(S_n^2) = E(\frac{1}{M} \sum_{i=0}^{M} U_{1n}^2 - \frac{1}{M(M+1)} (\sum_{j=1}^{n} X_j)^2) \tag{7.12}$$

$$= \frac{1}{M} E(\sum_{i=0}^{M} U_{in})^2 - \frac{1}{M(M+1)} (\sum_{j=1}^{n} \mu_j)^2 - \frac{1}{M(M+1)} \sum_{j=1}^{n} \sigma_j^2$$

substituting in (7.1) we get (7.11) □

From (7.11) we see that minimization of weighted flowtime with weights μ_j is equivalent to minimization of $\sum_{i=o}^{M} E(U_{in}^2)$, as in section 6. Formula (7.11) for deterministic jobs appears in Eastman, Even and Isaacs (1964).

8. General Weighted Flowtimes.

In this section we consider general jobs, and general weights. We derive an expression for the expected weighted flowtime. This generalizes (7.1), (6.1) and (5.5). It does not however share the simplicity of the previous formulas.

We start with the following useful decomposition.

Proposition 8.1: For any numbering of the jobs 1,...,n with fixed weights $W_1,\ldots,W_n$, and with random job completion times $C_1,\ldots,C_n$ we have, no matter what the strategy is:

$$\sum_{j=1}^{n} W_j C_j = \frac{W_n}{\mu_n} \sum_{j=1}^{n} \mu_j C_j + \sum_{k=1}^{n-1} \left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \sum_{j=1}^{k} \mu_j C_j \tag{8.1}$$

Proof: immediate. □

We will use (8.1) in particular when one or both of the following two conditions hold: The chosen order 1,...,n is such that the weight to expected processing time ratio is decreasing – $\frac{W_1}{\mu_1} \geq \ldots \geq \frac{W_n}{\mu_n}$ (in other words, 1,...,n is Smith's Rule ordering); the jobs are started in the order 1,...,n (in other words, SR is applied). The first condition ensures that $\left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \geq 0$ in (8.1). The second condition ensures that each of the summations $\sum_{j=1}^{k} \mu_j C_j$ is the weighted flowtime (with weights μ_j) of a subset of jobs $1,\ldots,k$ which are scheduled consecutively with no inserted idle times. Both conditions hold only if jobs are numbered by and scheduled according to Smith's Rule. Under the second condition the results of section 7 hold for any subset $\{1,\ldots,k\}$ and we have:

Theorem 8.2: If jobs are started in the order $1,\ldots, n$, then

$$E(\sum_{j=1}^{n} W_j C_j) = \frac{1}{M+1} E_1(\sum_{j=1}^{n} W_j \tilde{C}_j) + \frac{M}{2(M+1)} \sum_{j=1}^{n} W_j \mu_j (1 - \frac{\sigma_j^2}{\mu_j^2}) \tag{8.2}$$

$$+ \frac{M}{2} \sum_{j=1}^{n} \frac{W_j}{\mu_j} [E(S_j^{\;2}) - E(S_{j-1}^{\;2})]$$

Where $\tilde{C}_j$ are job completion times on a single machine, and E_1 is the expectation for scheduling weighted flowtime on a single machine.

Proof: Take expectations on (8.1), and substitute (7.1) to get:

$$E(\sum_{j=1}^{n} W_j C_j) = \frac{W_n}{\mu_n} [\frac{1}{2(M+1)} (\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^{\;2}$$

$$- \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 - \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2)]$$

$$+ \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}})[\frac{1}{2(M+1)}(\sum_{j=1}^{k} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{k} \mu_j^{\;2}$$

$$- \frac{M}{2(M+1)} \sum_{j=1}^{k} \sigma_j^2 - \frac{M}{2} S_0^2 + \frac{M}{2} E(S_k^2)] =$$

$$= \frac{W_n}{\mu_n} \frac{1}{M+1} E_1(\sum_{j=1}^{n} \mu_j \tilde{C}_j) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{1}{M+1} E_1(\sum_{j=1}^{k} \mu_j \tilde{C}_j)$$

$$+ \frac{W_n}{\mu_n} \frac{M}{2(M+1)} \sum_{j=1}^{n} (\mu_j^2 - \sigma_j^2) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2(M+1)} \sum_{j=1}^{k} (\mu_j^2 - \sigma_j^2)$$

$$- \frac{W_1}{\mu_1} \frac{M}{2} S_0^2 + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2} E(S_k^2) + \frac{W_n}{\mu_n} \frac{M}{2} E(S_n^2)$$

from which (8.2) follows. □

We note again the structure of the expected weighted flowtime on parallel machines: The first term is the expected value for a single, M+1 fold speed machine. The second is a delay per job created by the variance of the job's processing time. The third term is more awkward than is (5.5), it is a function of the sample variances of the remaining running times at each job start or completion; in (5.5) only S_0^2 and $E(S_n^2)$ appeared, now all the $E(S_j^2)$ are present.

9. Some general classes of strategies.

In sharp contrast to the deterministic case, for stochastic processing times there are several classes of strategies, which give different optimal solutions. We discuss these classes and their optimal solutions here. We start with definitions.

Π_{-1} - The optimal strategy in the class of strategies which pre-assign jobs to machines.

SR - Smith rule strategy.

Π_o - Some arbitrary list strategy.

Π_1 - The optimal list strategy; the class of list strategies includes all strategies which predetermine the order in which jobs get started, and jobs are then scheduled without idle times (work conserving).

Π_2 - The optimal dynamic work conserving strategy; optimal in the class of strategies which upon every job completion choose a job to start immediately, based on current state.

Π_3 - The optimal dynamic strategy with inserted idle time; a strategy in this class allows the insertion of idle time at each job completion, and allows the choice of the job to start to be delayed until the end of the idle time.

Π_4 - The optimal dynamic strategy when the actual value of each X_j is revealed when the processing of job j starts.

We briefly discuss Π_{-1} here. We are not aware of any previous work on this strategy. It is straightforward to see that

<u>Proposition 9.1</u>: The strategy Π_{-1} is the strategy which minimizes the deterministic problem with μ_j replacing X_j. The expected objective value of the former equals the objective value of the latter.

<u>Proof</u>: Let $J_i(1), \ldots, J_i(n_i)$ be the jobs assigned to machine i in the order in which they start. Then:

$$E\left(\sum_{j=1}^{n} W_j C_j\right) = E \sum_{i=0}^{M} \sum_{j=1}^{n_i} W_{J_i(j)} \sum_{k=1}^{j} X_{J_i(k)} \tag{9.1}$$

$$= \sum_{i=0}^{M} \sum_{j=1}^{n_i} W_{J_i(j)} \sum_{k=1}^{j} \mu_{J_i(k)}$$

which is the objective value of the deterministic problem with processing times μ_j. □

Hence, the problem of finding Π_{-1} for general expected weighted flowtime is NP-hard (Note! This is a stochastic problem which is NP-hard!). For the expected (unweighted) flowtime objective function, Π_{-1} is obtained by using the SPT schedule of deterministic jobs with processing times $\mu_1, \ldots, \mu_n$ to get $J_i(j)$, $j=1, \ldots, n_i$, $i=0, \ldots, M$. We have directly from (4.18):

<u>Proposition 9.2</u>: Let $\mu_1 \leq \ldots \leq \mu_n$, then for $S_0^2 = 0$,

$$E\left(\sum_{j=1}^{n} C_j \,\middle|\, \Pi_{-1}\right) = \frac{1}{M+1} \sum_{j=1}^{n} (n-j+1)\,\mu_j + \frac{1}{M+1} \sum_{i=0}^{M} i\, E(\tilde{U}_{in}) \tag{9.2}$$

where $\tilde{U}_{in}$ are the machine completion times $i = 0, \ldots, M$, ordered by: $E(\tilde{U}_{0n}) \leq \ldots \leq E(\tilde{U}_{Mn})$ (in contrast to our usual order of machine completion times $U_{0n} \leq \ldots \leq U_{Mn}$).

Proof: By theorem 9.1 we need to calculate the value for a deterministic problem, with processing times μ_j. (9.2) then follows from (4.18) □

For jobs with equal expected processing times we have, if $n = L(M+1) + K$, $0 \le K < M+1$, and $S_0^2 = 0$:

$$E(\sum_{j=1}^{n} C_j | \Pi_{-1}) = \frac{1}{M+1} \frac{n(n+1)}{2} \mu + \frac{1}{M+1} \frac{M(M+1)}{2} L \mu + \frac{1}{M+1} \frac{K(2M-k+1)}{2} \mu$$

and if n is a multiple of M+1,

$$E(\sum_{j=1}^{n} C_j | \Pi_{-1}) = \frac{n(n+1)}{2(M+1)} \mu + \frac{n\,M}{2(M+1)} \mu \qquad (9.3)$$

so, for jobs with equal expected processing times, using:

$$E(\sum_{j=1}^{n} C_j | \Pi_{-1}) - E(\sum_{j=1}^{n} C_j | \Pi_0) = \frac{n\,M}{2(M+1)} \frac{\sum_{j=1}^{n} \sigma_j^2}{n\,\mu} - \frac{M}{2} \frac{E(S_n^2 | \Pi_0)}{\mu} \qquad (9.4)$$

the difference is of order O(n) and is typical for strategies which preassign jobs to machines. The reason for this is that if jobs are preassigned then when one machine completes all its processing, other machines will typically still have $O(\sqrt{n})$ jobs to run.

Proposition 9.3: For minimization of expected flowtime, $E(\sum_{j=1}^{n} C_j | \Pi_{-1}) \ge E(\sum_{j=1}^{n} C_j | \text{SEPT})$.

Proof: Consider the following strategy: Apply Π_{-1} to the set of n jobs, and start jobs accordingly. Whenever a job completes, consider all the remaining unstarted jobs, apply Π_{-1} to the remaining jobs, and start the next job accordingly. This iterated procedure is clearly better than Π_{-1}. But, for the minimization of $E(\Sigma C_j)$ this iterated strategy is identical to SEPT. □

In general, for weighted flowtime Π_{-1} is NP-hard and comparison with SR or with any other list strategy is not clearcut. We have in general:

Proposition 9.4:

$$\left.\begin{matrix} E(\Sigma W_j C_j | \Pi_0) \\ E(\Sigma W_j C_j | \Pi_{SR}) \end{matrix}\right\} < \left.\begin{matrix} E(\Sigma W_j C_j | \Pi_{-1}) \\ E(\Sigma W_j C_j | \Pi_1) \end{matrix}\right\} < E(\Sigma W_j C_j | \Pi_2) < E(\Sigma W_j C_j | \Pi_3) < E(\Sigma W_j C_j | \Pi_4) \quad (9.4)$$

Proof: The proof of the inequality for Π_{-1} uses the same arguments as the proof of proposition 9.3. For all the other inequalities, the strategy on the right hand side of the inequality minimizes over a wider class of strategies than the one on the left hand side. □

The following proposition is required later:

Proposition 9.5: The strategy Π_4 does not use inserted idle time.

Proof: Without loss of generality we can assume that (whether idle time is used or not) whenever a job starts it starts on the available machine which has become available at the earliest time. Consider any nonrandomising schedule under the assumption that processing times of jobs become known at the instant in which their processing starts. Assume that at time t_o a machine becomes available, and all other machines are busy. Let $t_1, \ldots, t_m$ be the known times at which the other machines become available. Assume that idle time Δ is inserted and job j starts on the machine at $t_o + \Delta$; job j is the earliest job to start at time $\geq t_o$. Then Δ and j can be determined already at t_o, since no new information is obtained in $(t_o, t_o + \Delta)$. The machine will start job j at $t_o + \Delta$ and finish it at $t_o + \Delta + X_j$. The policy can now be modified as follows: Start job j at t_o; process it until $t_o + X_j$; then let the machine be idle until $t_o + X_j + \Delta$; release the value of X_j at $t_o + \Delta$; simulate the unmodified strategy otherwise. This is clearly a feasible schedule and it reduces C_j. Hence inserted idle time is unnecessary in Π_4. □

10. Bounds for Tails of Jobs

In the expressions for flowtime and weighted flowtime in Sections 5-8 the term $E(S_n^2)$, which is the expected value of the sample variance of the remaining processing times after U_{0n}, appears. We now discuss bounds on this term.

Theorem 10.1: Let $S_0^2 = 0$, and $X_1, \dots, X_n$ have distributions $F_1, \dots, F_n$. Let:

$$\bar{D}^2 = \max_{1 \le j \le n} \sup_{s > 0} \frac{1}{1-F_j(s)} \int_s^\infty (x-s)^2 \, dF_j(x) \qquad (10.1)$$

Then, for $k = 1, \dots, n$:

$$E(S_k^2) \le \frac{M}{M+1} \bar{D}^2 \qquad (10.2)$$

Proof: Consider any of the D_{ik}, $1 \le i \le M$, $1 \le k \le n$. D_{ik} is a remainder of a job that was started prior to U_{0k}, and has not completed its processing yet. If it is known which job D_{ik} consists of, and how long it has been processed prior to U_{0k}, then the distribution of D_{ik} is simply a tail of the distribution of this job. Condition on it being job j, and on job j having started at $U_{0k}-s$:

$$E(D_{ik}^2 | j,s) = E(D_{ik}^2 | D_{ik} = X_j - s,\ X_j > s) = \int_s^\infty (x-s)^2 \frac{dF_j(x)}{1-F_j(s)} \qquad (10.3)$$

Hence,

$$E(D_{ik}^2) \le \max_{1 \le j \le n} \max_{s > 0} \int_s^\infty (x-s)^2 dF_j(x)/(1-F(s)) = \bar{D}^2. \qquad (10.4)$$

Now,

$$S_k^2 = \frac{1}{M} \sum_{i=1}^M D_{ik}^2 - \frac{1}{M}\frac{1}{M+1} \left(\sum_{i=1}^M D_{ik} \right)^2$$

$$\le \frac{1}{M}\left(1 - \frac{1}{M+1}\right) \sum_{i=1}^M D_{ik}^2 = \frac{1}{M+1} \sum_{i=1}^M D_{ik}^2 \qquad (10.5)$$

and (10.2) follows. □

Clearly, $\bar{D}^2$ may by infinite for some distributions. We exclude such distributions. Also, one can choose $F_1,\ldots,F_n$ with $n \to \infty$ so that $\bar{D}^2 \to \infty$ with n. In many cases however, we can find simple bounds for the value of $\bar{D}^2$. We list some of these cases.

Case 1: $A \le X_j \le B$, the support of the distributions $F_1,\ldots,F_n$ is bounded by B. In this case, clearly $\bar{D}^2 \le B^2$.

Case 2: The distributions $F_1,\ldots,F_n$ satisfy, for every $s>0$:

$$E(X^2) \ge E((X-s)^2|X>s)$$

In this case:

$$\bar{D}^2 \le \max_{1\le j\le n} E(X_j^2) = \max_{1\le j\le n} (\mu_j^2+\sigma_j^2).$$

Case 3: X_j are NBU (new better than used). This implies Case 2, with the same bound.

Case 4: The hazard rate of each X_j is bounded from below.

$$h_j(t) = \frac{dF_j(t)}{1-F_j(t)} \ge \lambda_j$$

then $X_j-s|X_j>s$ is stochastically smaller than an $\exp(\lambda_j)$ random variable, so:
$$\bar{D}^2 \le \max_{1\le j\le n} \frac{2}{\lambda_j^2}.$$

11. Approximate Optimality of SR

We now prove the approximate optimality result:

Theorem: Let $\frac{W_1}{\mu_1} \ge \ldots \ge \frac{W_n}{\mu_n}$, and let $\bar{D}^2$ be defined as in section 10. Then:

$$E(\sum_{j=1}^{n} W_jC_j|SR) - E(\sum_{j=1}^{n} W_jC_j|\Pi_4) \le \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D}^2. \qquad (11.1)$$

Proof: By (8.1) we can rewrite the difference as:

$$\Delta = \frac{W_n}{\mu_n} \{E(\sum_{j=1}^{n} \mu_jC_j|SR) - E(\sum_{j=1}^{n} \mu_jC_j|\Pi_4)\} \qquad (11.2)$$

$$+ \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \{E(\sum_{j=1}^{k} \mu_jC_j|SR) - E(\sum_{j=1}^{k} \mu_jC_j|\Pi_4)\}$$

Under SR, for the problem of scheduling jobs $1,\dots,n$ with weights $W_1,\dots,W_n$, jobs are started in the order $1,\dots,n$. Hence, for everyy k, $E(\sum_{j=1}^{k} \mu_j C_j|SR)$ is the expected value of the nonpreemptive, no idle time list policy which starts the jobs in the order $1,\dots,k$.

The value of $E(\sum_{j=1}^{k} \mu_j C_j|\Pi_4)$ is more complicated. Here Π_4 schedules all the jobs $1,\dots,n$ to minimize $E(\sum_{j=1}^{n} W_j C_j)$ where no idle time is used (by Proposition 9.5), and where X_j becomes known when job j is started. If we now look at $\sum_{j=1}^{k} \mu_j C_j$ for the schedule given by Π_4, we may have jobs $1,\dots,k$ started in some order different from $1,\dots,k$ (and random), and with some of the jobs j, $j > k$, inserted in between. Let $\Pi_4(k)$ denote the optimal strategy for jobs $1,\dots,k$ with weights $\mu_1,\dots,\mu_k$. Clearly:

$$E(\sum_{j=1}^{k} \mu_j C_j|\Pi_4) \geq E(\sum \mu_j C_j|\Pi_4(k)) \tag{11.3}$$

Hence:

$$\Delta \leq \frac{W_n}{\mu_n}\{ E(\sum_{j=1}^{n} \mu_j C_j|SR) - E(\sum_{j=1}^{n} \mu_j C_j|\Pi_4) \} \tag{11.4}$$

$$+ \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \{ E(\sum_{J=1}^{k} \mu_j C_j|SR) - E(\sum_{j=1}^{k} \mu_j C_j|\Pi_4(k)) \}$$

since all the policies in (11.4) are now nonpreemptive and work conserving, by (7.10)

$$\Delta \leq \frac{W_n}{\mu_n} \frac{M}{2} E(S_n^2|SR) + \sum_{k=1}^{n-1} (\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}) \frac{M}{2} E(S_k^2|SR) \tag{11.5}$$

Bounding $E(S_k^2|SR)$ by (10.2)

$$\Delta \leq \frac{W_n}{\mu_n} \cdot \frac{M^2}{2(M+1)} \bar{D}^2 + \sum_{k=1}^{n-1} \left(\frac{W_k}{\mu_k} - \frac{W_{k+1}}{\mu_{k+1}}\right) \frac{M^2}{2(M+1)} \bar{D}^2$$

$$= \frac{M^2}{2(M+1)} \frac{W_1}{\mu_1} \bar{D} \qquad \square$$

12. Discussion.

In this paper we discussed the performance of Smith's Rule for nonpreemptive scheduling of a batch of jobs on parallel machines and have shown that its performance is very close to optimal. In the following discussion we briefly highlight the significance of our results, we list some possible extensions, and we explore some of the connections with other work in stochastic optimization.

Worst case performance of stochastic optization heuristics: Heuristics for deterministic combinatorial optimiization problems are assessed by considering worst case performance or average performance. The former is often much worse than the latter; however, for average performance one needs to assume a distribution on the population of all possible problems, which may not be acceptable. In stochastic optimization problems, a distribution on the population of possible problems is part of the model. Expected worst case performance is in fact an average over this distribution of problems. This may suggest that the use of heuristics for stochastic optimiization problems can be more successful than their use for deterministic problems. Our paper is a case in point for this.

Preemptive scheduling of a batch of jobs on parallel machines. Preemptive scheduling of a batch af jobs on a single machine, to minimize weighted flowtime, is optimized by using a Gittins index policy (Gittins 1979, 1982). On parallel machines this suggests to schedule at any moment the jobs with the highest Gittins index as a suboptimal heuristic, analogous to the use of Smith's Rule in the nonpreemptive case. It would be interesting to assess its performance.

Scheduling of a stream of arriving jobs. If jobs with various processing time distributions arrive at a single server in independent Poisson streams, then Smith's Rule and the Gittins index policy remain optimal (see Sevcik 1974, Klimov 1974, Harrison 1975, Meilijson and Weiss 1977, Gittins and Nash 1977). Using these rules for parallel servers provides suboptimal heuristics. Clearly, there is now a nonoptimal end effect at the end of each busy period; nevertheless it may be possible to bound the worst case behavior of these heuristics.

Extensions to control of queueing networks. In Weiss (1988) Gittins type priority rules for scheduling customers in a queueing network which is served by a single server (the server jumps between the nodes of the network and provides preemptive service) are derived. These may provide some heuristics for more conventional networks in which all the nodes are served simultaneously by several servers in parallel.

Restless Bandits. Whittle (1987) has recently considered some generalizations of Gittins' original Bandit process model. In scheduling terms these can be expressed as including several parallel servers as well as exogenous changes in waiting jobs. Whittle suggests a Gittins' type heuristic for these processes, and conjectures that under the appropriate asymptotic conditions these may converge to optimal. Our results in this paper provide a special case for which Whittle's conjecture holds.

Queueing network heuristics based on diffusion approximations. Recently Wein (1987) has derived some heuristics for the control of queueing networks by considering heavy traffic conditions and using diffusion approximations. Some parts of these heuristics appear to be priority type rules to schedule several types of customers. It is intriguing to try and find a possible connection between our current work, Whittle's conjecture, and Wein's results.

References:

1.Bruno, J., P. Downey and G. Frederickson, (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan, J. Assoc. Comput. Mach, 28, 100-113.

2.Coffman, E. G., Jr., M. Hofri, G. Weiss, (1987). Scheduling stochastic jobs with a two point distribution on two parallel machines. Submitted to Probab. Eng. Inform. Sci.

3.Cox, D. R., (1962). Renewal Theory, Methuen, London.

4.Cox, D. R. and H. D. Miller, (1965). Stochastic Processes, Methuen, London.

5.Eastman, W.L., Even, S., Isaacs, I.M. (1964) Bounds for the optimal scheduling of n jobs on m processors. Management Science 11, 268-279.

6.Feller, W., (1971). An Introduction to Probability Theory and Its Applications, Vol. II, 2nd Edition, Wiley, N.Y.

7.Frenk, J.B.G. and A.H.G. Rinnooy Kan (1987). The asymptotic optimality of the LPT rule. Math. Operat. Res. 12, 241-254.

8.Garey, M. R., D. S. Johnson (1979) Computers and Intractability: a Guide to the Theory of NP Completeness. Freeman, San Francisco.

9.Gittins, J. (1979) Bandit processes and dynamic allocation indices. J. Roy. Statist Soc. B. 14, 148-167.

10.Gittins, J. (1982) Forwards induction and dynamic allocation indices. In Deterministic And Stochastic Scheduling. (M.A.H Dempster et al, eds.) Reidel, Dordrecht, 125-156.

11.Gittins, J., P. Nash (1977) Scheduling, queues, and dynamic allocation indices. Proc. EMS, Prague, 1974. Czechoslovak Academy of Sciences, Prague, 191-202.

12.Glazebook, F. D. (1979) Scheduling tasks with exponential service times on parallel processors, J. Appl. Probab., 16, 685-689.

13.Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimaly. Oper. Res. 23, 270-282.

14.Kampke, T. (1985) On the optimality of static priority policies in stochastic scheduling on parallel machines, Oper. Res. (To Appear)

15.Klimov, G.P. (1974) Time-sharing service systems, I. Theory Probab. Appl. 19, 532-551.

16.Lenstra, J. K., A.H.G. Rinnooy-Kan, P. Brucker (1977) Complexity of machine scheduling problems, Ann. Discrete Math. 1, 343-362.

17.McNaughton, R. (1959) Scheduling with deadlines and loss functions, Management Science 6, 1-12.

18.Meilijson, I., G. Weiss, (1977) Multiple feedback at a single server station. Stochastic Processes Appl. 5, 195-205.

19.Pinedo, M., G. Weiss, (1979). Scheduling stochastic tasks on two parallel processors", Naval Research Logistics Quarterly, Vol. 26, No. 3, pp. 527-536.

20.Pinedo, M., G. Weiss (1987) The largest variance first policy in some stochastic scheduling problems, Oper. Res. 35, 5, Sept.-Oct.

21.Sevcik K.C. (1974) Scheduling for minimum total loss using service time distributions. J. Assoc. Comp. Mach. 21, 66-75.

22.Smith, W.E. (1956) Various optimizers for single stage production, Naval Research Logistics Quarterly, 3, 59-66.

23.Van der Wal, J., Hordijk, A. (1987) Personal communication, OberWolfach meeting on Operations Research, February.

24.Weber, R. R., (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time, J. Appl. Probab., 19, 167-182.

25.Weber, R. R., P. Varaiya and J. Walrand, (1986) Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime, J. Appl. Probab. 23.

26.Wein, L.M. (1987) Asymptotically optimal scheduling of a two station multiclass queueing network. PhD Dissertation, Stanford University.

27.Weiss, G. (1988) Branching bandit processes. Probability in Engineering and Informational Sciences, to appear.

28.Weiss, G., C. C. Huang, C. L. Li, S. Liu, M. Pinedo, J. Song (1987) A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime. Georgia Tech ISyE technical report submitted to OR letters.

29. Weiss, G. and M. Pinedo, (1980) Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions, J. Appl. Probab., 17, 187-202.

30. Whittle, P. (1987) Restless Bandits. Talk presented at the GMOOR meeting, Passau, West Germany.

Appendix Proof of Theorem 5.1.

In this appendix we prove Theorem 5.1 about the convergence of the Markov chain $D_{1m},\ldots,D_{Mm}$, for F nonarithmetic. The proof closely follows similar derivations in Feller (1971). We assume that the iid processing times distribution F is a nonarithmetic distribution. We let $B = \sup\{x|F(x) < \infty\} \le \infty$, and we let $\Omega = \{y_1,\ldots, y_M | 0 \le y_1 \le y_2 \le \ldots \le y_M \le B\} \subseteq R^M$ be the sample space for $D_{1m},\ldots, D_{Mm}$; we say I is an open interval in Ω if $I \subseteq \Omega$ is of the form $(\alpha_1,\beta_1)\times\ldots\times(\alpha_m,\beta_m)$ and say $|I| < \epsilon$ if $|\beta_i-\alpha_i| < \epsilon$, $i= 1,\ldots, M$, and $|I| > \epsilon$ if $|\beta_i-\alpha_i| > \epsilon$, $i=1,\ldots,M$. Let $\gamma_m(d\underline{y})$, $m = 0,1,\ldots$ denote the probability distribution of $D_{1m},\ldots,D_{Mm}$, and let $K(\underline{x},d\underline{y})$ denote the transition kernel of the chain, so that (Feller, Chapter VIII, Section 7, p. 270.)

$$\gamma_{m+1}(d\underline{y}) = \int K(\underline{x},d\underline{y})\ \gamma_m(d\underline{x}), \tag{A.1}$$

describes the one step Markov transitions, and let $K^{(n)}(\underline{x},d\underline{y})$ be the n step transition kernel, from γ_m to γ_{m+n}.

For our process the one step transition is given by:

$$\begin{aligned}\gamma_{m+1}(dy_1,\ldots, dy_M) = &\int_0^\infty \gamma_m(d(y_1+u),\ldots,d(y_M+u))\ dF(u)\\ &+ \int_0^\infty \gamma_m(du,d(y_2+u),\ldots,d(y_M+u))\ dF(y_1+u) \qquad \text{(A.2)}\\ &\ \vdots\\ &+ \int_0^\infty \gamma_m(du,d(y_1+u),\ldots,d(y_{j-1}+u),d(y_{j+1}+u),\ldots,d(y_M+u))\ dF(y_j+u)\\ &\ \vdots\\ &+ \int_0^\infty \gamma_m(du,d(y_1+u),\ldots,d(y_{M-1}+u))\ dF(y_M+u).\end{aligned}$$

for $y_1 \le \ldots \le y_M$.

We recall some of the definitions from Feller:

Definition A.1: (Feller, Chapter VI, section II, Definition 2, p. 207)

A measure α is called a stationary measure for the kernel K if $\alpha = \alpha_o = \alpha_1 = \dots \alpha_m = \alpha_{m+1} = \dots$ in the transition relation (A.1).

Definition A.2: (Feller, Chapter VIII, section 7, Definition 1, p. 271)

A measure α is strictly positive in Ω if $\alpha(I) > o$ for every open interval in Ω.

Definition A.3: (Feller, ditto, definition 3) The kernel K is ergodic if there exists a strictly positive probability distribution α on Ω such that $\gamma_n \to \alpha$ independently of the initial probability γ_o.

Dual to (A.1) we also look at the relation:

$$U_{n+1}(\underline{x}) = \int K(\underline{x}, d\underline{y})\, U_n(\underline{y}) \tag{A.3}$$

which for our process has the form:

$$\begin{aligned} U_{n+1}(x_1, \dots, x_M) = & \int_o^{x_1} U_n(x_1-u, x_2-u, \dots, x_M-u)\, dF(u) \qquad \text{(A.4)} \\ & + \int_{x_1}^{x_2} U_n(u-x_1, x_2-x_1, \dots, x_M-x_1)\, dF(u) \\ & \vdots \\ & + \int_{x_j}^{x_{j+1}} U_n(x_2-x_1, \dots, x_j-x_1, u-x_1, x_{j+1}-x_1, \dots, x_M-x_1) dF(u) \\ & \vdots \\ & + \int_{x_M}^{\infty} U_n(x_2-x_1, \dots, x_M-x_1, u-x_1)\, dF(u). \end{aligned}$$

for $x_1 \leq \dots \leq x_M$.

Definition A.4: (Feller, ditto, definition 3) The kernel K is regular if whenever U_o is uniformly continuous (so that for every $\epsilon > 0$ there exists $\delta > 0$ such that if $\|\underline{x}'-\underline{x}''\| < \delta$, $\underline{x}',\underline{x}'' \in \Omega$ then $|U_n(\underline{x}')-U_n(\underline{x}'')| < \epsilon$) the whole family of functions U_n, $n = 0,1,\ldots$ defined by A.3 is equicontinuous (so that for every $\epsilon > 0$ there exists $\delta > 0$ such that if $\|\underline{x}'-\underline{x}''\| < \delta$, $\underline{x}',\underline{x}'' \in \Omega$, then $|U_n(\underline{x}')-U_n(\underline{x}'')| < \epsilon$).

Definition A.5: (Feller, ditto, definition 2) The kernel K is strictly positive if $K(\underline{x},I) > 0$ for every $\underline{x} \in \Omega$ and every open interval $I \subseteq \Omega$.

To these we add:

Definition A.6: The kernel K is asymptotically strictly positive if for every $\epsilon > 0$ there exists N such that for all $n > N$, $K^{(n)}(\underline{x},I) > 0$ for all $\underline{x} \in \Omega$ which satisfies $0 < x_i < 1/\epsilon$, and all open intervals I, $I \subseteq \Omega$ which satisfy $|I| > \epsilon$, and which are contained in $(0,1/\epsilon)^M$.

We want to prove that the chain $D_{1m},\ldots,D_{Mm}$ is ergodic, with $\alpha(d\underline{x}) = \frac{M!}{\mu^M} \bar{F}(x_1)\ldots \bar{F}(x_M)\, d\underline{x}$ for $\underline{x} \in \Omega$ as the limiting distribution. The proof is in four steps.

- α is a stationary distribution for the kernel K.
- The kernel K is regular.
- The kernel K is asymptotically strictly positive.
- The above three points imply ergodicity

The last point is an extension of Feller, ditto, theorem 2, p. 272 in which asymptotically strictly positive replaces strictly positive.

Proposition A.1: The measure $\alpha(d\underline{y}) = \frac{M!}{\mu^M} \bar{F}(y_1)\ldots \bar{F}(y_M)dy_1\ldots dy_M$ for $\underline{y} \in \Omega$ is a stationary measure for the chain $D_{1m},\ldots,D_{Mm}$, $m = 0,1,\ldots$.

Proof: Substitute in the right hand side of (A.2) to obtain, for $y_1 \leq \dots \leq y_M$:

$$\left(\frac{M!}{\mu^M} dy_1 \dots dy_M\right) \times \int_0^\infty \bar{F}(y_1+u) \dots \bar{F}(y_M+u)\, dF(u)$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_2+u) \dots \bar{F}(y_M+u)\, dF(y_1+u)$$

$$\vdots$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_1+u) \dots \bar{F}(y_{j-1}+u)\, \bar{F}(y_{j+1}+u) \dots \bar{F}(y_M+u)\, dF(y_j+u)$$

$$\vdots$$

$$+ \int_0^\infty \bar{F}(u)\, \bar{F}(y_1+u) \dots \bar{F}(y_{M-1}+u)\, dF(y_M+u) =$$

$$= \frac{M!}{\mu^M} dy_1 \dots dy_M \int_0^\infty \frac{d}{du} \left(\bar{F}(u)\bar{F}(y_1+u) \dots \bar{F}(y_M+u)\right) du$$

$$= \frac{M!}{\mu^M} dy_1 \dots dy_M\, \bar{F}(y_1) \dots \bar{F}(y_M). \qquad \square$$

Proposition A.2: The kernel K described by (A.2) is regular.

Proof: The proof for two machines (M = 1) follows Feller (ditto, example a, p. 272). Assume for given $\epsilon > 0$ there is $\delta < 0$ such that if $\|x'-x''\| < \delta$ than $|U_m(x')-U_m(x'')| < \epsilon$. We now show that this implies (for the same ϵ): $|U_{m+1}(x')-U_{m-1}(x'')| < \epsilon$. Assume $x' < x''$.

$$|U_{m+1}(x')-U_{m+1}(x'')| \leq \int_0^{x'} |U_m(x'-y) - U_m(x''-y)|\, dF(y)$$

$$+ \int_{x'}^{x''} |U_m(y-x') - U_m(x''-y)|\, dF(y)$$

$$+ \int_{x''}^\infty |U_m(y-x') - U_m(y-x'')|\, dF(y)$$

Note that for $x' \leq y \leq x''$, $|(y-x') - (x''-y)| \leq |y-x'| + |x''-y| = |x''-x'| < \delta$ and of course $|(x'-y) - (x''-y)| < \delta$, $|(y-x') - (y-x'')| < \delta$, so that the whole expression is $\leq \int \epsilon \, dF(y) = \epsilon$.

This proof fails for $M > 1$, since the induction step involves $U_m(y-x_1', x_2'-x_1', \ldots) - U_m(y-x_1'', x_2''-x_1'', \ldots)$ and the distance in norm between the arguments may double.

However, the proposition still holds, by a different proof. For $\epsilon > 0$ let $\delta < 0$ satisfy: $\|\underline{x}'-\underline{x}''\| < \delta$ implies $|U_o(\underline{x}') - U_o(\underline{x}'')| < \epsilon$. Use now $\delta/2$, and take any $\underline{x}'$, $\underline{x}''$ such that $\|\underline{x}'-\underline{x}''\| < \delta/2$. Consider $U_n(\underline{x}')$, $U_n(\underline{x}'')$. They equal the values of U_o evaluated at the points reached from $\underline{x}'$, $\underline{x}''$ after n transitions. Let $X_1, \ldots, X_n$ drawn independently from F be the random variables which define the n step transition (the processing times). Condition on $X_1 = y_1, \ldots, X_n = y_n$; define $\tilde{\underline{x}}_o' = (0, x_1', \ldots, x_n')$, $\tilde{\underline{x}}_o'' = (0, x_1'', \ldots, x_n'')$ and let $\tilde{\underline{x}}_{i+1}'$, $\tilde{\underline{x}}_{i+1}''$ be defined inductively as the vectors obtained by adding y_{i+1} to the smallest components of $\tilde{\underline{x}}_i'$, $\tilde{\underline{x}}_i''$ and reordering the M+1 components. Then $\|\tilde{\underline{x}}_n' - \tilde{\underline{x}}_n''\| < \delta/2$. Obtain $\underline{x}_n'$, $\underline{x}_n''$ by subtracting the first component from the last M components of $\tilde{\underline{x}}_n'$, $\tilde{\underline{x}}_n''$. Then $\|\underline{x}_n'-\underline{x}_n''\| < \delta$. Hence, $|U_o(\underline{x}_n') - U_o(\underline{x}_n'')| < \epsilon$. But $U_o(\underline{x}_n') = U_n(\underline{x}'|\underline{y})$, $U_o(\underline{x}_n'') = U_n(\underline{x}''|\underline{y})$. Taking expectations, we get $|U_n(\underline{x}') - U_n(\underline{x}'')| < \epsilon$. □

<u>Proposition A.3</u>: The kernel K is asymptotically strictly positive.

<u>Proof</u>: We give first the proof for 2 machines, $M = 1$. We proceed similarly to Feller (1971, Chapter V, Section 4a, pp. 147-148). Recall that y is a point of increase of a distribution F if $F\{I\} > 0$ for every open interval I containing y; a point of increase of $K(x,dy)$ and of the n stage transition $K^{(n)}(x,dy)$ is defined similarly. Let $\epsilon > 0$ be given. We start by choosing $b \geq \min(B, 1/\epsilon)$ such that b is a point of increase of

F (for B finite take b = B). We take a fixed arbitrary x, $x < \min(b, 1/\epsilon) \leq b$. Clearly x is (the only) point of increase of $K^{(0)}(x,dy)$ and b–x is a point of increase of $K^{(1)}(x,dy)$. Also, if $y \leq b$ is a point of increase of $K^{(n)}(x,dy)$, then y is also a point of increase of $K^{(n+2)}(x,dy)$. Hence, x (b–x) is a point of increase of $K^{(n)}(x,dy)$ for all n which are even (odd).

Since F is nonarithmetic we can find a point of increase of F, say a, such that $0 < kb-\ell a = h < \epsilon$, with $k+\ell$ even. By considering $x + X_{i_1} + \ldots + X_{i_\ell} - X_{j_1} - \ldots - X_{j_k}$ we see that x–h is a point of increase of $K^{(k+\ell)}(x,dy)$, and of $K^{(n)}(x,dy)$ for n even, $n \geq (k+\ell)$. Similarly, considering $x+X_{j_1} + \ldots + X_{j_k} - X_{i_1} - \ldots - X_{i_\ell}$, we see that x+h is a point of increase of $K^{(k+\ell)}(x,dy)$, and of $K^{(n)}(x,dy)$ for n even, $n \geq (k+\ell)$. In the same way, (b–x) – h and (b–x) + h are points of increase of $K^{(n)}(x,dy)$ for all n odd, $n \geq k+\ell+1$. Next one sees that for n even, $n \geq 2(k+\ell)$, all of $0 < x-2h, x-h, x, x+h, x+2h < b$ are points of increase of $K^{(n)}(x,dy)$, with similar statement for b–x. Take $N = \lceil \frac{b}{h} \rceil (k+\ell)$; then for n even, $n \geq N$, all points of the form $0 < x \pm mh < b$ are points of increase of $K^{(n)}(x,dy)$ while for n odd, $n \geq N$, all points of the form $0 < b-x \pm mh < b$ are points of increase of $K^{(n)}(x,dy)$. Thus for $n \geq N$, every interval I of length $\geq \epsilon > h$ within $[0,\bar{b}]$ contains at least one point of increase of $K^{(n)}(x,dy)$, and has $K^{(n)}(x,I) > 0$, as required.

To extend the proof to M+1 machines, take some given $n_1, \ldots, n_M$, $y_1, \ldots y_M$, and $u_1 \ldots u_M$ such that u_i is a point of increase of $K^{(n_i)}(y_i,dy)$, for the two machine transition kernel. Then for the M+1 machine transition kernel $K(\underline{x},d\underline{z})$, $(u_1, u_1+u_2, \ldots, u_1+u_2+\ldots+u_M)$ is a point of increase for $K^{(n_1+\ldots+n_M)}(y_1, y_1+y_2, \ldots, y_1+\ldots+y_M, d\underline{z})$. This is shown by induction: Assuming $\underline{z}^{(j-1)} = (u_1, u_1+u_2, \ldots, u_1+\ldots+u_{j-1}, u_1+\ldots+u_{j-1}+y_j, \ldots, u_1+\ldots+u_{j-1}+y_j+\ldots+y_M)$ is a point of increase of $K^{(n_1+\ldots+n_{j-1})}(y_1, y_1+y_2, \ldots, y_1+\ldots+y_M, d\underline{z})$. Then the n_j step transition from $\underline{z}^{(j-1)}$ has $\underline{z}^{(j)} = (u_1, u_1+u_2, \ldots, u_1+\ldots+u_{j-1}, u_1+\ldots+u_j, u_1+\ldots+u_j+y_{j+1}, \ldots)$ as a point of increase. This is all that is necessary to go from 2 to M+1 machines. □

<u>Proposition A.4</u>: Propositions A1, A2, A3 imply ergodicity.

<u>Proof</u>: This differs from Feller's theorem 2 (ditto) in that strictly positive is replaced by asymptotically strictly positive, and we prove it for R^M. Let then α denote the stationary measure. Let E denote expectation with respect to α. Let $U_o \in C\,[-\infty,\infty]$ (continuous with limits at $\pm\infty$, hence uniformly continuous). By stationarity $E(U_o) = E(U_1)\ldots$. Also, $E|U_k|$ decrease with k, since:

$$E|U_k| = \int \left| \int U_{k-1}(y)\, K(x,dy) \right| d\alpha(x) \le \int\int |U_{k-1}(y)|\, K(x,dy)\, d\alpha(x) = E|U_{k-1}| \qquad \text{(A.8)}$$

so $\lim_{k\to\infty} E|U_k| = m$ exists. If U_o is uniformly continuous this implies that U_k are equicontinuous (proposition A.2) and so (by the selection theorem, see Feller, ditto, chapter VIII section 6 theorem 3) a convergent subsequence $U_{n_k} \to V_o$ exists. Applying the transition N times $U_{n_k+N} \to V_N$. By dominated (or bounded) convergence, $E(U_{n_k}) \to E(V_o)$, and, $E|U_{n_k}| \to E|V_o|$, similarly for V_N, and so:

$$E(V_N) = E(V_o) = E(U_o),\ E|V_N| = E|V_o| = m.$$

We now show that this implies that V_o cannot change sign. By the definition of the stationary measure, similar to (A.8), $E|V_N| = E|V_O|$ is equivalent to:

$$\int \left|\int V_0(y)\ K^{(N)}(x,dy)\right| \alpha(dx) = \int\int |V_0(y)|\ K^{(N)}(x,dy)\ \alpha(dx). \quad \text{(A.9)}$$

Assume that V_0 does change sign. Then by continuity (V_0 is continuous since $U_n \epsilon C[-\infty,\infty]$) we can find $\epsilon > 0$ and two open intervals $I_1, I_2 \subseteq \Omega \cap (0,1/\epsilon)^M$, such that $|I_1| > \epsilon$, $|I_2| > \epsilon$ and $V_0 > 0$ on I_1, $V_0 < 0$ on I_2. By asymptotic strict positivity of K, we can find N such that for all $x \in \Omega \cap (0,1/\epsilon)^M$, $K^{(N)}(x,I_1) > 0$ and $K^{(N)}(x,I_2) > 0$. But this contradicts (A.9). In particular, if $E(U_o) = 0$ then $V_o \equiv 0$, and (by considering $U_0(x)-m$) in general, $V_o(x) \equiv m$. Hence, V_o is constant, and so are of course $V_1, V_2, \ldots$. This limit is independent of the subsequence, and so $U_n(x) \to E(U_o)$. In other words, if for any x we look at the sequence of distributions $k^{(n)}(x,dy)$, and take expectation, denoting the expectation by E_n, we have $E_n(U_o) \to E(U_o)$ for all $U_o \in C[-\infty,\infty]$. But this implies $K^{(n)}(x,dy) \to \alpha(dy)$ at all points of continuity of α. □

Industrial and Systems Engineering

Report Series No. J-88-9

May, 1988

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

*This research was supported in part by NSF grant ECS-8712798.

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss*
School of Industrial and systems Engineering
Georgia Institute of Technology

Abstract

We consider scheduling a batch of jobs with stochastic processing times on parallel machines, with minimization of expected weighted flowtime as objective. Smith's Rule, which orders job starts by decreasing ratio of weight to expected processing time provides a natural heuristic for this problem. We show that it has a turnpike optimality property: the expected number of optimal decisions which are not according to Smith's Rule is bounded by a constant independent of the number of jobs.

*This research was supported in part by NSF grant ECS-8712798.

TURNPIKE OPTIMALITY OF SMITH'S RULE IN PARALLEL MACHINES STOCHASTIC SCHEDULING

Gideon Weiss
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

1. INTRODUCTION

A batch of n jobs (customers, tasks) is to be processed by $M+1$ identical parallel machines (servers, processors). Job j requires processing time X_j, to be provided by any one of the machines, where the value of X_j is specified by a probability distribution F_j, and the X_j's are drawn indepent of each other and of the schedule; a weight w_j is associated (as holding cost per unit time) with job j. Under some arbitrary scheduling rule let C_j be the completion time of job j; the cost of the schedule under this rule is the (random) weighted flowtime $\sum_{j=1}^{n} w_j C_j$.

The general problem of minimizing the expected weighted flowtime is intractable. A simple plausible heuristic is provided by Smith's Rule - order the job starts by decreasing weight to expected processing time ratio.

A large body of research has been devoted to this problem and its various special cases - we shall give a brief survey of the results presently. Notwithstanding all these results, we contend that in a very general framework Smith's Rule provides a satisfactory solution to the problem. In a recent paper (Weiss 1988) we have shown that the additional expected cost of using Smith's Rule, above the optimal cost, is negligible. In the present paper we prove a turnpike optimality result by showing that the expected number of times that it is optimal to start a job not according to Smith's Rule is negligible (for an earlier turnpike result, see Coffman, Hofri and Weiss 1988).

It is known that SEPT (Shortest Expected Processing Time First) and SR (Smith's Rule) minimize expected flowtime and weighted flowtime respectively, on a single machine (Smith 1956, Conway, Maxwell and Miller 1967). For parallel machines, SEPT minimizes expected flowtime in many important special cases, notably when for all i,j X_i and X_j are stochasticaly comparable (that is either $X_i \leq_{ST} X_j$ or $X_j \leq_{ST} X_i$, alternatively, either $F_i(x) \geq F_j(x)$ for all x or $F_j(x) \geq F_i(x)$ for all x - Weber, Varaiya and Walrand 1986). Deterministic processing times, exponentially distributed processing times, and processing times which are distributed like tails of a single increasing hazard rate distribution are among the special cases of this remarkable result (McNaughton 1959, Bruno, Downey and Frederickson 1981, Glazebrook 1979, Weiss and Pinedo 1981, Weber 1982). In general however, SEPT may fail to be optimal (see Pinedo and Weiss 1987, Coffman, Hofri and Weiss 1988). Minimization of weighted flowtime on parallel machines, for deterministic processing times X_j and general weights w_j, is an NP-hard combinatorial optimization problem (Lenstra, Rinnooy Kan and Brooker 1977, Garey and Johnson 1979), and very little can be said about optimal strategies for the minimization of expected weighted flowtime in the stochastic case (Kampke 1986).

The use of Smith's Rule as a heuristic can be quite inefficient - a worst case analysis shows that for deterministic processing times the ratio $R = (\Sigma\, w_jC_j \mid SR) / (\Sigma\, w_jC_j \mid OPT)$ can be as high as 1.20 (Weiss et al 1988). This worst case involves a very large number of very short jobs and ≈ 0.3 M very long jobs.

The following intuitive discussion captures the essence of the problem: The argument in favour of SEPT (as well as for Smith's Rule), is that at the beginning of the schedule there is a large number of jobs waiting and SEPT (or Smith's Rule) tend to reduce the number of jobs (their cost rate) fastest. This argument suffices to prove optimality for a single processor, and it applies to parallel processors as well. For parallel processors there exists, however, a counter argument: Towards the end of the schedule, as jobs are completed, there are no more new jobs to start and the processors fall idle one after the other; this means that processing at the end becomes inefficient and this of course has an effect on the objective function. Thus it seems that one ought to try to reduce these inefficiency periods. Minimization of these periods is hard (in the deterministic case, for two machines, it is equivalent to minimizing the makespan which is NP-hard), and it is not achieved by SEPT. If anything, it is asymptotically best to use LEPT to minimize makespan and the inefficiency periods (see Frenk and Rinnooy Kan, 1987). Nevertheless this inefficiency at the end is a boundary effect and is of marginal value - in particular it does not grow with the number of jobs n, so long as jobs remain uniformly bounded in some sense. It appears that this end effect is the only counter indication against optimality of SEPT and of Smith's Rule. By quantifying this end effect we get bounds on the expected difference in the objective function value between Smith's Rule and the optimal strategy, and bounds on the expected number of jobs which the optimal strategy does not schedule according to Smith's Rule.

2. FORMULATION AND RESULTS

For jobs $j = 1,\dots,n$ we assume the processing times X_j are nonnegative random variables drawn from distributions F_j. We let $\bar{F}_j = 1 - F_j$ and assume F_j have finite means μ_j and variances σ_j^2.

We require the following quantities, defined in terms of the distributions:

$$\bar{D}^2 = \max_{1\le j\le n} \sup \left\{ \int_s^\infty (x-v)^2 \, dF_j(x) \,/\, \bar{F}_j(v) \;\middle|\; v \ge 0,\ \bar{F}_j(v) > 0 \right\} \qquad (1)$$

$$\delta^{(M)} = \min_{j_0,\dots,j_M} \inf\left\{ \int_0^\infty \bar{F}_{j_0}(x+v_0) \cdot\ldots\cdot \bar{F}_{j_M}(x+v_M)\, dx \,/\, \bar{F}_{j_0}(v_0) \cdot\ldots\cdot \bar{F}_{j_M}(v_M) \;\middle|\; v_0,\dots,v_M \ge 0,\ \bar{F}_{j_0}(v_0) > 0,\ \dots,\ \bar{F}_{j_M}(v_M) > 0 \right\} \qquad (2)$$

roughly these measure respectively how long and how short remainders of jobs can be. In addition we need the following quantities which depend also on the weights:

$$\left(\frac{w}{\mu}\right)_{\max} = \max_{1\le j\le n} \frac{w_j}{\mu_j} \qquad (3)$$

$$\mu_{\min} = \min_{1\le j\le n} \mu_j \qquad (4)$$

$$\left(\Delta\frac{w}{\mu}\right)_{\min} = \min_{i\ne j} \left| \frac{w_i}{\mu_i} - \frac{w_j}{\mu_j} \right| \qquad (5)$$

we assume that $\bar{D}^2 < \infty$ and $\delta^{(M)} > 0$.

We will consider two classes of strategies: Π_0 includes all nonpreemptive, work conserving (i.e. use no inserted idle time), nonrandomizing strategies which base decisions at time t on the history of the schedule up to time t. Π includes all nonpreemptive strategies which allow insertion of idle time as well as randomization and which base decisions at time t not only on the history of the schedule up to time t but also on the actual realized values of remaining processing times and inserted idle periods which occupy the machines at time t. In other words, when using strategies in Π, the processing times of jobs become known at the moment that their processing starts; similarly, the lengths of inserted idle periods are known at their start.

Assume now that M+1 machines are available to process jobs 1,...,n starting at time 0. In a previous paper we have shown:

<u>Theorem 1: (Weiss 1988)</u> For any strategy $\pi \in \Pi$

$$E(\Sigma w_j C_j \mid SR) - E(\Sigma w_j C_j \mid \pi) \leq \frac{M^2}{2(M+1)} (\frac{w}{\mu})_{max} \bar{D}^2 \quad (6)$$

In this paper we will show:

<u>Theorem 2:</u> For any strategy $\pi \in \Pi_0$ such that

$$E(\Sigma w_j C_j \mid SR) - E(\Sigma w_j C_j \mid \pi) \geq 0 \quad (7)$$

let L be the number of times that π starts a job not according to Smith's Rule. Then

$$E(L \mid \pi) \leq \frac{M^2}{2} (\frac{w}{\mu})_{max} \bar{D}^2 / (\Delta\frac{w}{\mu})_{min} \mu_{min} \delta^{(M)} . \quad (8)$$

For practical purposes one is usually interested only in the smaller class of strategies, Π_0. However, we need Theorem 1 to hold for Π in order to prove Theorem 2 for Π_0. We do not know whether Theorem 2 holds for Π, it is quite plausible that it does not. We take a close look at the two classes of strategies in Section 4.

There exist many practical scenarios under which the values of $\bar{D}^2$ and of $\delta^{(M)}$, remain bounded for any number of jobs n. These include the following:

(a) If all the jobs belong to a finite number of types K, summarized by processing time distributions $F_1,\ldots,F_K$, and if $\bar{D}^2 < \infty$ and $\delta^{(M)} > 0$ for a set of jobs which includes M+1 jobs of each type, then $\bar{D}^2$ and $\delta^{(M)}$ are finite and remain fixed as n grows.

(b) If the hazard rate functions of the processing time distributions are all uniformly bounded by some upper bound value $\bar{\lambda}$ and some lower bound value $\underline{\lambda}$, then $\bar{D}^2 \leq \frac{2}{\underline{\lambda}^2}$, and $\delta^{(M)} \geq \frac{1}{(M+1)\bar{\lambda}}$.

(c) If all the jobs have NBU (New Better than Used) type processing time distributions, and μ_j, σ_j^2 are uniformly bounded then $\bar{D}^2$ is bounded by a uniform bound on $\mu_j^2 + \sigma_j^2$.

(d) If the processing times have lattice probability distributions with a common time unit τ then $\delta^{(M)} \geq \tau$.

In all these special cases it is still possible to have jobs which are long or short without any bound; the bounds are only on the probabilities.

If $\bar{D}^2$ and $(\frac{w}{\mu})_{max}$ have fixed upper bounds, and if $\delta^{(M)}$, $(\Delta\frac{w}{\mu})_{min}$, and μ_{min} have fixed lower bounds, independent of the number of jobs n, then as we let $n \longrightarrow \infty$, we will have $E(L) / n \approx O(\frac{1}{n})$ and $E(\Sigma w_j C_j | SR) / E(\Sigma w_j C_j | \pi) \approx 1 + O(\frac{1}{n^2})$.

3. APPROXIMATE OPTIMALITY OF SMITH'S RULE

In this section we summarize some of the results about expected weighted flowtime and prove Theorem 1. For more details and additional results see Weiss, 1988.

Consider the schedule obtained by starting the jobs in the order $I(1),\dots,I(n)$, without preemptions and without inserted idle time. Let $U_{00} \le \dots \le U_{M0}$ denote the ordered times at which the $M+1$ machines become available initialy - we shall assume throughout that $\sum_{i=0}^{M} U_{i0} = 0$; for Theorems 1 and 2 we shall also require that $U_{i0} = 0$, $i = 0,\dots,M$. For $j = 1,\dots,n$ let $U_{0j} \le \dots \le U_{Mj}$ be the ordered times when the machines become available after the completion of jobs $I(1),\dots,I(j)$; the starting time of job $I(j)$ (the j'th job to start) is U_{0j-1}. We have the recursion:

$$(U_{0j},\dots,U_{Mj}) = \text{Order Statistics of } (U_{0j-1}+X_{I(j)},U_{1j-1},\dots,U_{Mj-1}) \qquad (9)$$

For $j=0,\dots,n$, denote $D_{ij} = U_{ij} - U_{0j}$, $i = 1,\dots M$. Denote by S_j^2 the sample variance of $U_{0j},\dots,U_{Mj}$ (or of $0,D_{1j},\dots,D_{Mj}$)

$$S_j^2 = \frac{1}{M}\sum_{i=1}^{M} D_{ij}^2 - \frac{1}{M(M+1)}\Big(\sum_{i=1}^{M} D_{ij}\Big)^2. \qquad (10)$$

It is easily seen that the completion of job $I(j)$ is at

$$C_{I(j)} = \frac{1}{M+1}\sum_{k=1}^{j} X_{I(k)} + \frac{M}{M+1} X_{I(j)} - \frac{1}{M+1}\sum_{i=1}^{M} D_{ij-1}. \qquad (11)$$

Directly from (11) we get:

<u>Lemma 1:</u> The weighted flowtime can be decomposed as:

$$\sum_{j=1}^{n} w_j C_j = \frac{1}{M+1}\sum_{j=1}^{n}\Big(\sum_{k=j}^{n} w_{I(k)}\Big) X_{I(j)} + \frac{1}{M+1}\sum_{j=1}^{n} w_{I(j)}\Big(M\, X_{I(j)} - \sum_{i=1}^{M} D_{ij-1}\Big) \qquad (12)$$

By examining the recursive relations between the D_{ij}'s the following key formula emerges:

Lemma 2: The job remainders are linked by the relation:

$$2 \sum_{j=1}^{n} \sum_{i=1}^{M} X_{I(j)} D_{ij-1} = M \sum_{j=1}^{n} X_j^2 + M(M+1) S_0^2 - M(M+1) S_n^2 \qquad (13)$$

Proof: For two machines we have $D_{1j} = |X_{I(j)} - D_{1j-1}|$; squaring and adding over j gives (13) for $M=1$. For $M>1$, (13) is obtained by applying the argument to each pair of machines, and summing over all pairs. □

We now consider the special case of $w_j = \mu_j$. Taking expectations over (12) and (13), we have:

Lemma 3: For any nonpreemptive work conserving strategy $\pi \in \Pi$,

$$E(\sum_{j=1}^{n} \mu_j C_j | \pi) = \frac{1}{2(M+1)} (\sum_{j=1}^{n} \mu_j)^2 + \frac{1}{2} \sum_{j=1}^{n} \mu_j^2 - \frac{M}{2(M+1)} \sum_{j=1}^{n} \sigma_j^2 \qquad (14)$$
$$- \frac{M}{2} S_0^2 + \frac{M}{2} E(S_n^2 | \pi)$$

Proof: The main step in the proof of Lemma 3 is to note that

$$E(X_{I(j)} D_{ij-1}) = E(D_{ij-1} E(X_{I(j)} | D_{ij-1}) = \qquad (15)$$
$$= E(D_{ij-1} E(E(X_{I(j)} | I(j)) | D_{ij-1}) =$$
$$= E(D_{ij-1} \mu_{I(j)})$$

The second equality holds since D_{ij-1} is a function of $X_{I(1)},\ldots,X_{I(j-1)}$ only, and so $E(X_{I(j)} | I(j), D_{ij-1}) = E(X_{I(j)} | I(j))$.□

The remarkable thing about formula (14) is that the only term which depends on the schedule at all is the last term $E(S_n^2)$; furthermore, the rest of the expression depends only on the first two moments of the distributions $F_1,\ldots,F_n$.

A direct corollary to Lemma 3 is:

<u>Corollary 4:</u> Let $\pi, \pi' \in \Pi$ be two strategies, and asssume that π is work conserving. Assume also that $S_0^2 = 0$. Then

$$E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi'\Big) \le \frac{M}{2} E(S_n^2 \mid \pi) \le \frac{M^2}{2(M+1)} \bar{D}^2 \qquad (16)$$

<u>Proof:</u> By Lemma 5 (in the next section) it is possible to construct a strategy $\tilde{\pi}'$ which is randomizing but does not use inserted idle time, such that $I(1),\dots,I(n)$ have the same distribution under $\tilde{\pi}'$ as under π', and

$$E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \tilde{\pi}'\Big) \le E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi'\Big)$$

Hence

$$E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi'\Big) \le E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi\Big) - E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \tilde{\pi}'\Big)$$

$$= \frac{M}{2} E(S_n^2 \mid \pi) - \frac{M}{2} E(S_n^2 \mid \tilde{\pi}') \le \frac{M}{2} E(S_n^2 \mid \pi) \qquad (17)$$

By definition (1) and the assumption that $S_0^2 = 0$, we get $E(D_{ij}^2) \le \bar{D}^2$ for all i,j which by definition (10) implies the second inequality in (16). □

From this point onwards we will assume that the jobs are ordered by Smith's Rule, that is $\frac{w_1}{\mu_1} \ge \dots \ge \frac{w_n}{\mu_n}$.

For any strategy we can rewrite the weighted flowtime as:

$$\sum_{j=1}^{n} w_j C_j = \frac{w_n}{\mu_n} \sum_{j=1}^{n} \mu_j C_j + \sum_{k=1}^{n-1} \Big(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\Big) \sum_{j=1}^{k} \mu_j C_j \qquad (18)$$

It is now easy to obtain:

<u>Proof of Theorem 1:</u> We use (18) to write:

$$\Delta = E\Big(\sum_{j=1}^{n} w_j C_j \mid SR\Big) - E\Big(\sum_{j=1}^{n} w_j C_j \mid \pi\Big) \qquad (19)$$

$$= \frac{w_n}{\mu_n} \Big\{E\Big(\sum_{j=1}^{n} \mu_j C_j \mid SR\Big) - E\Big(\sum_{j=1}^{n} \mu_j C_j \mid \pi\Big)\Big\} +$$

$$\sum_{k=1}^{n-1} \Big(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\Big) \Big\{E\Big(\sum_{j=1}^{k} \mu_j C_j \mid SR\Big) - E\Big(\sum_{j=1}^{k} \mu_j C_j \mid \pi\Big)\Big\}$$

We note that $\sum_{j=1}^{k} \mu_j C_j | SR$ and $\sum_{j=1}^{k} \mu_j C_j | \pi$ are the weighted flowtimes (with weights μ_j), of jobs $1,\ldots,k$, which are scheduled by SR in that order with no inserted idle time. Under π they are scheduled not necessarily in the order $1,\ldots,k$, and the schedule may include inserted idle periods, which consist of those jobs $j > k$ which π has started before the start of job k, as well as additional idle periods inserted by π prior to the start of job k. Nevertheless, the scheduling strategy provided by π for jobs $1,\ldots,k$ is in Π. Since SR is work conserving for every set $1,\ldots,k$, we can apply Corollary 4, and, noting that $\frac{w_n}{\mu_n} \geq 0$ and $\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}} \geq 0$, we get:

$$\Delta \leq \left\{ \frac{w_n}{\mu_n} + \sum_{k=1}^{n-1} \left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}} \right) \right\} \frac{M^2}{2(M+1)} \bar{D}^2 \tag{20}$$

which collapses to (6). □

4. DISCUSSION OF STRATEGIES

We take a closer look at strategies, starting with the wider and more complicated class of strategies Π. Consider a strategy $\pi \in \Pi$. To describe π we need to describe how a schedule is constructed by π. The construction of a schedule by π consists of a sequence of decisions on job or idle period starts, and updates of state. We use the following state desctiption: $s = (t, \underline{d}, N, H)$ where t is the time; $\underline{d} = d_0, \dots, d_M$ is the machine availability vector, where $d_i = 0$ indicates that machine i is available at t while $d_i > 0$ indicates that machine i is not available at time t and will only become available at time $t + d_i$ - recall that strategies in Π can make use of the values of remaining processing times, in other words, the values d_i are deterministic and known; N is the set of as yet unstarted jobs, on which the current information is summarized by F_j the distribution of X_j for $j \in N$; Finally, H is the history of the schedule up to time t, which includes the start times of the jobs and the allocations of the machines over the period prior to t and also any additional information that π may use, excluding predictive information about the X_j, $j \in N$ - in particular H may include the information which is used by π to randomize decisions.

The initial state is $s = (t_0, \underline{d}, N, H)$ where t_0 is the schedule start time, $\underline{d}$ are the initial machine availabilties, with $d_i = 0$ for at least one machine, $N = \{1, \dots, n\}$ includes all the jobs, and H is empty. At a decision time let the state be s, and at least one machine is available, with i_0 the available machine with lowest index. The decision will make machine i_0 unavailable for a time τ, to be determined by the decision. Based on the state s, and possibly using randomization, π will choose a job to start from N, or will choose an idle period. If an idle period is chosen, then τ will be the length of this idle period, which will be generated from some distribution G determined by π according to s. If job $j \in N$ is chosen to start then $\tau = X_j$ is generated from F_j. In either case, the value of τ becomes known immediately. Following the decision by π the state s is updated: let $\Delta t = \min(\tau, d_i, i \neq i_0)$, then $t := t + \Delta t$; $d_i := d_i - \Delta t$, $i \neq i_0$, $d_{i_0} := d_{i_0} + \tau - \Delta t$; $N := N - \{j\}$ if job j was chosen, $N := N$ if an idle period was chosen; finaly H is updated to indicate the allocation of machine i_0 and the job start if a job was started, and new information regarding future randomization may be added.

We shall add three provisos on decisions:

- If all the machines are available, a job has to start.

- We assume that for any state s, $\tau > 0$, and the time until the next decision, Δt, satisfies $E(\Delta t \mid \Delta t > 0) \geq \delta^{(M)}$ (in the case of randomization this has some complicated implications on G - however, we shall have no trouble verifying the condition for the schedules which we use in the proof).

- We assume that if $N = \phi$, an idle period of length $\tau = \max (d_0, \ldots, d_M)$ is inserted.

The schedule is complete if N is empty and all machines are available. The proviso that not all the machines can be idle assures that the length of the schedule is no more than $\sum_{j=1}^{n} X_j$ and therefore has finite expectation. The lower limit on $E(\Delta t \mid \Delta t > 0)$, assures that the number of decisions taken to construct the schedule has a finite expectation. The third proviso makes the schedule complete at the completion of the last job.

The class of strategies Π_0 is much simpler than Π. The description is similar except that all the features regarding randomization and insertion of idle time are excluded.

The next Lemma on the construction of strategies with no idle times is crucial to the proof of Theorems 1 and 2.

Lemma 5: Let $\pi' \in \Pi$ be an arbitrary strategy which uses inserted idle time, then there exists a strategy $\pi \in \Pi$ such that π is randomizing but does not use inserted idle time, and such that the order in which jobs are started, $I(1),\ldots,I(n)$, has the same probability distribution under π' and π. Also, for all $j=1,\ldots,n$, $E(C_j|\pi) \leq E(C_j|\pi')$.

Proof: We prove the Lemma by describing the strategy π. We describe π by showing how a schedule is constructed by π. In addition to the state of the schedule constructed by π, which we denote by $s = (t,\underline{d},N,H)$, we shall make use of an auxilliary state, $s' = (t',\underline{d}',N',H')$, which will simulate the schedule constructed by π'. Initially we will have $t = t' = t_0$, and $s = s'$. Subsequently we shall use π' to update s' and use s' to update s. Starting from the initial states at time t_0 we will make a sequence of decisions and updates, in such a way that both s' and s get updated at every decision, and we have always $t' \geq t$, $N' = N$, and both $\underline{d}$ and $\underline{d}'$ have at least one available machine. Assume that following a sequence of decisions we have states s, s'. We use s' to obtain the decision taken by π', we generate a machine time τ, and we update s' accordingly. We then update s: If the decision of π' is to start an idle period, the state s remains unchanged. If the decision of π' is to start job $j \in N'$, we start job j on the machine with lowest index available in $\underline{d}$, we use the same value of X_j as realized in the update of s', that is we use a machine period $\tau = \tau'$, and we update s accordingly.

We note the following: Whenever s is updated, it is by assigning a job to an available machine; hence the schedule described by the state s uses no inserted idle time. At a decision state s, a sequence of one or more decisions which are taken by π' determines which job is chosen to start out of N; these decisions depend on the state s and the auxilliary state s' but are otherwise nonpredictive, thus the choice of job j is randomized but nonpredictive; the assumption of the lower bound $\delta^{(M)}$ assures that the randomization process has a finite expected number of steps. Once the job j is chosen, its processing time X_j is generated by F_j in the update of s. We let the state s describe the schedule generated by π; we have seen so far that π is a randomizing strategy using no inserted idle time, for scheduling the jobs $1,\ldots,n$. If we think of s' as part of the state history H, then it is clear that $\pi \in \Pi$. It is also clear from the construction that s' is a state which also describes a schedule, namely a schedule constructed by π'. Moreover, for every realization of this construction, the order in which jobs start, $I(1),\ldots,I(n)$, is the same for the two schedules described by s and by s'; this proves the statement that the job starting orders have the same distributions under both strategies. It remains to show that $E(C_j | \pi) \leq E(C_j | \pi')$. At any state s define $\underline{u}$ as the ordered vector of $t+d_0, t+d_1, \ldots, t+d_M$, and define $\underline{u}'$ analogously for s'. Initially, at $t=t'=t_0$, $\underline{u} = \underline{u}'$. We claim that thereafter, after each joint update of s' and s, $\underline{u} \leq \underline{u}'$. this is seen inductively: If π' chooses inserted idle time, only $\underline{u}'$ changes, and it has its smallest component increased and then all its components reordered; this can clearly not reduce any component; if π' chooses a job, the same τ is added to the smallest component of both $\underline{u}$ and $\underline{u}'$, and then they are both reordered, but then $\underline{u}'$ dominates $\underline{u}$ before reordering, and therefore also after the reordering. Since $\underline{u}'$ dominates $\underline{u}$ at every decision, job j starts earlier in s then in s', and, a-fortiori $E(C_j | \pi) \leq E(C_j | \pi')$. □

Immediately from Lemma 5, we have:

Corollary 6: The optimal schedule in Π does not use inserted idle time.

Corollary 7: Allowing randomization in Π or in Π_0 does not improve the optimal schedule.

Proof: Since the optimal policies use no idle time, the number of possible actions at a decision moment is at most n, and there are n decisions to be taken in the whole construction of the schedule. The proof is by backwards induction. □

5. TURNPIKE OPTIMALITY OF SMITH'S RULE:

The role of $\bar{D}^2$ and of $\delta^{(M)}$:

In the proof of Corollary 4 we saw that under every work conserving strategy π, if $S_0^{\;2} = 0$ then for all j, $E(S_j^{\;2} \mid \pi) \le \frac{M}{M+1}\bar{D}^2$; thus $\bar{D}^2$ puts an upper bound on the expected remaining processing times at each decision moment. The inequality (16) which is proved in corollary 4 was used in the proof of Theorem 1 and will also be used in the proof of Theorem 2.

On the other hand, consider the j'th decision moment of a strategy $\pi \in \Pi_0$ at which job ℓ starts its processing, and assume that all the other processors are occupied, so $D_{i\,j-1} > 0$, $i=1,\dots,M$. Then directly from the definition

$$E(\ \min(\ X_\ell,\ D_{1\,j-1},\ \dots,\ D_{Mj-1}\ ))\ \ge \delta^{(M)} \tag{21}$$

This puts a lower limit on the expected time between decisions, for every $\pi \in \Pi_0$. This is required in the proof of Theorem 2, but not in the proof of Theorem 1.

Proof of Theorem 2:

Let $\pi \in \Pi_0$ be a strategy which outperforms SR so that (7) holds. Rewriting (19) we have:

$$\begin{aligned} 0 &\ge E(\sum_{j=1}^{n} w_j C_j \mid \pi) - E(\sum_{j=1}^{n} w_j C_j \mid SR) \\ &= \frac{w_n}{\mu_n}\{E(\sum_{j=1}^{n} \mu_j C_j \mid \pi) - E(\sum_{j=1}^{n} \mu_j C_j \mid SR)\} + \\ &\sum_{k=1}^{n-1} (\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\ \{E(\sum_{j=1}^{k} \mu_j C_j \mid \pi) - E(\sum_{j=1}^{k} \mu_j C_j \mid SR)\} \end{aligned} \tag{22}$$

Consider the jobs $1,\dots,k$ whose completion times appear in $E(\sum_{j=1}^{k} \mu C_j \mid \pi)$. The strategy π provides a srategy in Π for scheduling these jobs which possibly uses some inserted idle times in the form of processing times X_ℓ where job ℓ is started by π before job k, and $\ell > k$; π also uses randomization in scheduling jobs $1,\dots,k$, since their order may be determined by the values of the inserted jobs ℓ. For $k = 1,\dots,n-1$, let $\tilde{\pi}_{(k)}$ be the randomizing work conserving strategy for scheduling jobs $1,\dots,k$ which is constructed in Lemma 5 from π. We rewrite (22)

$$0 \ge \frac{w_n}{\mu_n}\{E(\sum_{j=1}^{n}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{n}\mu_j C_j \mid SR)\} + \tag{23}$$
$$\sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)}) - E(\sum_{j=1}^{k}\mu_j C_j \mid SR)\} +$$
$$\sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)})\}$$

We now apply corollary 4 to $\tilde{\pi}_{(k)}$ and obtain a bound on the first two summands as in the proof of Theorem 1. This gives us the following inequality which must hold for any π which outperforms SR:

$$\frac{M^2}{2(M+1)}(\frac{w}{\mu})_{max}\bar{\bar{D}}^2 \ge \sum_{k=1}^{n-1}(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}})\{E(\sum_{j=1}^{k}\mu_j C_j \mid \pi) - E(\sum_{j=1}^{k}\mu_j C_j \mid \tilde{\pi}_{(k)})\} \tag{24}$$

We now obtain a lower bound on the difference between π and $\tilde{\pi}_{(k)}$.

Denote by $I_{\ell k}$ the indicator of the event that π starts job ℓ before job k - recall that SR is characterized by $I_{\ell k} = 0$ whenever $\ell > k$. Condition now on $I_{\ell k} = 1$ for some $\ell > k$, and on the state of the schedule constructed by π, at the instant that the service of job ℓ starts. Let $d_1,\dots,d_M$ be the availablility vector of all the other machines, it consists of remaining processing times of jobs which have had some known amount of service, and of the full processing time of jobs which start simultaneously with job ℓ. Let $\tau = \min(X_\ell, d_1,\dots,d_M)$.

Since $\pi \in \Pi_0$ is not predictive, the remaining processing times are distributed like $F_j(x+v)/\bar{F}_j(v)$ where v is the amount of processing already received by job j, and so the time until the earliest next job start has expected value of at least $\delta^{(M)}$. Removing the processing of job ℓ from the schedule will not increase any job completion time within jobs $1,\ldots,k$, and at least one job (either k or some job k' with $k' < k$) will start earlier by τ. So the expected saving in the weighted flowtime (with weights μ_j) conditional on $I_{\ell k} = 1$, where $\ell > k$, is at least $\mu_{min}\ \delta^{(M)}$.

The saving by removing a job ℓ which π has inserted in front of job k appears to count only once for every k. However, it is easy to construct $\tilde{\pi}_{(k)}$ in several stages, starting from the schedule under π, and at each successive stage removing the last inserted idle job. Each such a removal will reduce the expected weighted flowtime of the sum over $1,\ldots,k$ by the above amount. Note however, that if several idle jobs were inserted simultaneously, the removal of all of them from the schedule may have the same effect as to remove only one. So we can only assume an improvement of $\mu_{min}\ \delta^{(M)} / (M+1)$ in the expected sum over $1,\ldots,k$ for each event $I_{\ell k} = 1$, where $\ell > k$.

We have thus shown:

$$E\left(\sum_{j=1}^{k} \mu_j C_j \mid \pi\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \tilde{\pi}_{(k)}\right) \geq \sum_{\ell>k} P(I_{\ell k} = 1)\ \mu_{min}\ \delta^{(M)}/(M+1) \tag{25}$$

Using the lower bound on $\left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right)$, and summing over $k=1,\ldots,n-1$, we have:

$$\sum_{k=1}^{n-1} \left(\frac{w_k}{\mu_k} - \frac{w_{k+1}}{\mu_{k+1}}\right) \left\{E\left(\sum_{j=1}^{k} \mu_j C_j \mid \pi\right) - E\left(\sum_{j=1}^{k} \mu_j C_j \mid \tilde{\pi}_{(k)}\right)\right\} \geq \tag{26}$$

$$\sum_{k=1}^{n-1} \sum_{\ell>k} P(I_{\ell k} = 1)\ \left(\Delta\frac{w}{\mu}\right)_{min}\ \mu_{min}\delta^{(M)}/\ (M+1)$$

But

$$E(L \mid \pi) = \sum_{k=1}^{n-1} \sum_{\ell>k} P(I_{\ell k} = 1) \tag{27}$$

and substituing (26), (27) in (24) the theorem follows. □

6. DISCUSSION

We discuss briefly our results, their significance, and possible directions for future research.

Worst case performance of stochastic optimization heuristics. Heuristics for deterministic combinatorial optimization problems are assessed by considering worst case performance or average performance. The former is often much worse than the latter; however, for average performance one needs to assume a distribution on the population of all possible problems, which may not be acceptable. In stochastic optimization problems, a distribution on the population of possible problems is part of the model. Expected worst case performance is in fact an average over this distribution of problems. This may suggest that the use of heuristics for stochastic optimization problems can be more successful than their use for deterministic problems. Our paper is a case in point for this.

Almost sure bounds. We have shown approximate optimality and turnpike optimality of Smith's Rule in minimizing weighted flowtime, in a very general setting. Without further assumptions on the processing time distributions, these results seem the best possible. Under more specific model assumptions it may however be possible to obtain stronger results, such as almost sure bounds on the difference in objective values and on the number of non SR optimal decisions, rather than bounds on expectations.

An explicit formula for expected weighted flowtime. Our results are based on various formulas which we derived for expected weighted flowtime. The most general form is obtained by substituting (14) into (18). We have, if jobs are started in the order $1,2,\ldots,n$:

$$E\Big(\sum_{j=1}^{n} w_j C_j \;\Big|\; M+1 \text{ machines}\Big) = \frac{1}{M+1} E\Big(\sum_{j=1}^{n} w_j C_j \;\Big|\; \text{One machine}\Big) + \frac{M}{2(M+1)} \sum_{j=1}^{n} w_j \mu_j \Big(1 - \frac{\mu_j^2}{\sigma_j^2}\Big) + \frac{M}{2} \sum_{j=1}^{n} \frac{w_j}{\mu_j} \{E(S_j^2) - E(S_{j-1}^2)\} \qquad (28)$$

The first term is the weighted flowtime on a single, $M+1$ fold speed machine; it depends only on the first moments of the processing time distributions, it is of order $O(n^2)$, and it is minimized by SR. The second term is composed of per job delays caused by the parallel processing; it is a function of the means and the variances of the processing times, it is of order $O(n)$, and it is independent of the schedule. The last term is the intractable part which may depends on the full description of the processing time distributions as well as on the schedule. For large n it may however exhibit some limiting steady state behaviour.

Preemptive scheduling of a batch of jobs on parallel machines. Preemptive scheduling of a batch af jobs on a single machine, to minimize weighted flowtime, is optimized by using a Gittins index policy (Gittins 1979, 1982). On parallel machines this suggests to schedule at any moment the jobs with the highest Gittins index as a suboptimal heuristic, analogous to the use of Smith's Rule in the nonpreemptive case. A very special case of i.i.d jobs with a two point distribution on two parallel machines has been analyzed by Coffman, Hofri and Weiss 1987. It would be interesting to generalize the results.

Scheduling of a stream of arriving jobs. If jobs with various processing time distributions arrive at a single server in independent Poisson streams, then Smith's Rule and the Gittins index policy remain optimal (see Sevcik 1974, Klimov 1974, Harrison 1975, Meilijson and Weiss 1977, Gittins and Nash 1977). Using these rules for parallel servers provides suboptimal heuristics. Clearly, there is now a nonoptimal end effect at the end of each busy period; nevertheless it may be possible to bound the worst case behavior of these heuristics.

Extensions to control of queueing networks. In Weiss (1988) Gittins type priority rules for scheduling customers in a queueing network which is served by a single server (the server jumps between the nodes of the network and provides preemptive service) are derived. These may provide some heuristics for more conventional networks in which all the nodes are served simultaneously by several servers in parallel.

Restless Bandits. Whittle (1987) has recently considered some generalizations of Gittins' original Bandit process model. In scheduling terms these can be expressed as including several parallel servers as well as exogenous changes in waiting jobs. Whittle suggests a Gittins' type heuristic for these processes, and conjectures that under the appropriate asymptotic conditions these may converge to optimal. Our results in this paper provide a special case for which Whittle's conjecture holds.

Queueing network heuristics based on diffusion approximations. Recently Wein (1987) has derived some heuristics for the control of queueing networks by considering heavy traffic conditions and using diffusion approximations. Some parts of these heuristics appear to be priority type rules to schedule several types of customers. It is intriguing to try and find a possible connection between our current work, Whittle's conjecture, and Wein's results.

References:

1.Bruno, J., P. Downey and G. Frederickson, (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan, J. Assoc. Comput. Mach, 28, 100-113.

2.Coffman, E. G., Jr., M. Hofri, G. Weiss, (1987). Scheduling stochastic jobs with a two point distribution on two parallel machines. Probab. Eng. Inform. Sci. To appear.

3.Conway R.W., Maxwell W.L., Miller L.W. (1967) Theory of scheduling Addison Wesley,Reading MA.

4.Frenk, J.B.G. and A.H.G. Rinnooy Kan (1987). The asymptotic optimality of the LPT rule. Math. Operat. Res. 12, 241-254.

5.Garey, M. R., D. S. Johnson (1979) Computers and Intractability: a Guide to the Theory of NP Completeness. Freeman, San Francisco.

6.Gittins, J. (1979) Bandit processes and dynamic allocation indices. J. Roy. Statist Soc. B. 14, 148-167.

7.Gittins, J. (1982) Forwards induction and dynamic allocation indices. In Deterministic And Stochastic Scheduling. (M.A.H Dempster et al, eds.) Reidel, Dordrecht, 125-156.

8.Gittins, J., P. Nash (1977) Scheduling, queues, and dynamic allocation indices. Proc. EMS, Prague, 1974. Czechoslovak Academy of Sciences, Prague, 191-202.

9.Glazebook, F. D. (1979) Scheduling tasks with exponential service times on parallel processors, J. Appl. Probab., 16, 685-689.

10.Harrison, J.M. (1975) Dynamic scheduling of a multiclass queue: discount optimaly. Oper. Res. 23, 270-282.

11.Kampke, T. (1985) On the optimality of static priority policies in stochastic scheduling on parallel machines, Oper. Res. (To Appear)

12.Klimov, G.P. (1974) Time-sharing service systems, I. Theory Probab. Appl. 19, 532-551.

13.Lenstra, J. K., A.H.G. Rinnooy-Kan, P. Brucker (1977) Complexity of machine scheduling problems, Ann. Discrete Math. 1, 343-362.

14.McNaughton, R. (1959) Scheduling with deadlines and loss functions, Management Science 6, 1-12.

15.Meilijson, I., G. Weiss, (1977) Multiple feedback at a single server station. Stochastic Processes Appl. 5, 195-205.

16.Pinedo, M., G. Weiss (1987) The largest variance first policy in some stochastic scheduling problems, Oper. Res. 35, 5, Sept.-Oct.

17.Sevcik K.C. (1974) Scheduling for minimum total loss using service time distributions. J. Assoc. Comp. Mach. 21, 66-75.

18.Smith, W.E. (1956) Various optimizers for single stage production, Naval Research Logistics Quarterly, 3, 59-66.

19.Weber, R. R., (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flow time, J. Appl. Probab., 19, 167-182.

20.Weber, R. R., P. Varaiya and J. Walrand, (1986) Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime, J. Appl. Probab. 23.

21.Wein, L.M. (1987) Asymptotically optimal scheduling of a two station multiclass queueing network. PhD Dissertation, Stanford University.

22.Weiss, G. (1988) Branching bandit processes. Probab. Eng. Inform. Sci., to appear.

23.Weiss, G., C. C. Huang, C. L. Li, S. Liu, M. Pinedo, J. Song (1987) A worst case analysis of Smith's rule for scheduling parallel machines to minimize weighted flowtime. Georgia Tech ISyE technical report no. J-87-12, submitted to OR letters.

24.Weiss, G. (1988) Approximation results in parallel machines stochastic scheduling. Georgia Tech ISyE technical report no. J-88-3.

25.Weiss, G. and M. Pinedo, (1980) Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions, J. Appl. Probab., 17, 187-202.

26.Whittle, P. (1987) Restless Bandits: Activity allocation in a changing world. Adv. appl. Probab. to appear.