GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

Date: August 13, 1979

Project Title: Finitely Convergent Cutting Plane Methods for Solving the Quadratic Assignment Problem

Project No: E-24-603 Green card NSF

Project Director: Dr. Mokhtar S. Bazaraa

Sponsor: National Science Foundation

Agreement Period: From 9/1/79 Until 2/28/81
(12 month budget period plus 6 month unfunded flexibility perio

Type Agreement: Grant No. ENG-7908375; dated 7/12/79

Amount: $29,539 NSF
13,124 GIT (E-24-338)
$42,663 TOTAL

Reports Required: Final Project Report

Sponsor Contact Person (s):

Technical Matters

William L. Brogan
Program Director
Systems Theory and Operations Research
Division of Engineering
National Science Foundation
Washington, D. C. 20550
(202) 632-5881

Contractual Matters
(thru OCA)

Mr. Philip M. King
Grants Administrator
MPE/STIA Branch
National Science Foundation
Washington, D. C. 20550
(202) 632-5965

Defense Priority Rating: None

Assigned to: Industrial & Systems Engineering (School/Laboratory)

COPIES TO:

- Project Director
- Division Chief (EES)
- School/Laboratory Director
- Dean/Director—EES
- Accounting Office
- Procurement Office
- Security Coordinator (OCA)
- ✓ Reports Coordinator (OCA) Rodgers
- Library, Technical Reports Section
- Office of Computing Services
- Director, Physical Plant
- EES Information Office
- Project File (OCA)
- Project Code (GTRI)
- Other OCA Research Property Coordinator

CA—3 (3/76)

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION SHEET

Date 11/18/81

Project Title: Finitely Convergent Cutting Plane Methods for Solving the Quadratic Assignment Problem

Project No: E-24-603

Project Director: Dr. Mokhtar S. Bazaraa

Sponsor: National Science Foundation

Effective Termination Date: 2/28/82 *

Clearance of Accounting Charges: 2/28/82 *

Grant/Contract Closeout Actions Remaining:

- [ ] Final Invoice and Closing Documents
- [x] Final Fiscal Report (FCTR)
- [x] Final Report of Inventions (if positive)
- [ ] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other ______

* Work completed and final report submitted within the six (6) month unfunded flexibility period.

Assigned to: ISyE (School/Laboratory)

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting
Procurement/EES Supply Services
Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)
Library
EES Public Relations (2)
Computer Input
Project File
Other ______

FORM OCA 10:781

E-21-667

NATIONAL SCIENCE FOUNDATION
Washington, D.C. 20550

FINAL PROJECT REPORT
NSF FORM 98A

PLEASE READ INSTRUCTIONS ON REVERSE BEFORE COMPLETING

PART I–PROJECT IDENTIFICATION INFORMATION

1. Institution and Address	2. NSF Program	3. NSF Award Number
Georgia Institute of Technology Atlanta, Georgia 30332	Systems Theory and Operations Research	ENG-7908375
	4. Award Period From 9/1/79 To 2/28/82	5. Cumulative Award Amount $60,266

6. Project Title

Finitely Convergent Cutting Plane Methods for Solving the Quadratic Assignment Problem

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

This study is concerned with the development of several exact and heuristic methods for solving the quadratic assignment problem. Two approaches are investigated, namely, cutting planes and branch and bound. First, using a suitable formulation, the problem is transformed into the minimization of a concave quadratic objective function over the assignment polytope. Several cutting plane procedures are devised. These methods delete local optimal solutions whose objective value do not improve the incumbent solution. Second, a branch and bound procedure that uses symmetric properties of the problem is developed. Through the use of computational expedients, both the cutting plane and branch bound algorithms are transformed into efficient heuristics for solving the quadratic assignment problem.

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
Mokhtar S. Bazaraa		11/11/81

TABLE OF CONTENTS

FINITELY CONVERGENT CUTTING PLANE METHODS
FOR SOLVING THE QUADRATIC ASSIGNMENT PROBLEM

ABSTRACT

This study is concerned with the development of several exact and heuristic methods for solving the quadratic assignment problem. Two approaches are investigated, namely, cutting planes and branch and bound. First, using a suitable formulation, the problem is transformed into the minimization of a concave quadratic objective function over the assignment polytope. Several cutting plane procedures are devised. These methods delete local optimal solutions whose objective value do not improve the incumbent solution. Second, a branch and bound procedure that uses symmetric properties of the problem is developed. Through the use of computational expedients, both the cutting plane and branch and bound algorithms are transformed into efficient heuristics for solving the quadratic assignment problem.

1. INTRODUCTION

The quadratic assignment problem, as given by Koopmans and Beckmann, can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} f_{ik} d_{j\ell} x_{ij} x_{k\ell}$$

$$\text{Subject to} \quad \sum_{i=1}^{m} x_{ij} = 1 \qquad j=1,\ldots,m$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i=1,\ldots,m$$

$$x_{ij} = 0 \text{ or } 1 \qquad i,j=1,\ldots,m.$$

The problem involves assigning m indivisible facilities to m locations. The flow between objects i and k is f_{ik} and the distance between locations j and ℓ is $d_{j\ell}$. The objective is to assign the facilities to the locations in such a way that the sum of pairwise interactions among objects weighed by the distance between their respective locations is minimized.

There exists two approaches for solving the quadratic assignment problem exactly. The first approach utilizes the concept of branch and bound or implicit enumeration. Secondly, through an appropriate transformation, the problem can be reformulated as a linear mixed-integer program which is solved by cutting planes or by a suitable mixed-integer programming package.

Due to the complexity of the quadratic assignment problem, in general, the above exact methods cannot solve problems with dimension $m > 15$ effectively. Thus for larger problems, a considerable amount of effort has been given to the development of inexact methods that obtain good quality solutions with a reasonable computational effort. Inexact methods for solving the quadratic assignment problem fall under one of the following classifications.

a. Construction Methods

Starting with a partial solution or the null assignment, a complete assignment is reached by iteratively locating one or more objects at each iteration.

b. Improvement Methods

Starting with a complete assignment of objects, an improvement over the incumbent is sought by interchanging the locations of several objects. The procedure is terminated when no further improvements are possible.

c. Hybrid Methods

Methods in this class combine several features of exact and inexact methods.

In this research, several exact and hybrid methods are developed for solving the quadratic assignment problem. Particularly, several cutting planes and branch and bound algorithms are devised for providing exact solutions to the problem. Computational expedients and heuristics are incorporated in these algorithms, resulting in efficient hybrid methods. These methods produced the best known solutions for standard problems in the literature, and in some cases, produced improved solutions.

This report gives a brief summary of the algorithms. A detailed description of the methods and the computational results is given in the Appendix which contains three research manuscripts.

2. CUTTING PLANE METHODS

Denote the feasible region of the problem by the set X_A and denote the associated assignment polytope by the set X. The quadratic assignment problem can be formulated as follows:

$$\text{Minimize} \quad x^t Sx$$
$$\text{Subject to} \quad x \epsilon X_A$$

where S is a suitable symmetric matrix and the superscript t denotes the transpose operation.

The formulation adopted here transforms the above problem into a concave quadratic program. Observe that if we replace the objective function with $x^t Dx = x^t[S - IM]x$ where M is a constant, then the problem is essentially unchanged since $x^t IMx = mM$, a constant. Moreover, if one selects M to be larger than the greatest row sum of S, then one can show that D is negative definite. Since the minimum of a strictly concave function over a bounded polyhedral set occurs at an extreme point, the problem can be written as:

$$\text{Minimize} \quad x^t Dx$$
$$\text{Subject to} \quad x \epsilon X .$$

A local optimal to the above problem needs not be globally optimal. However, cutting planes can be used to find the global optimal solution among local optima.

2.1. General Approach

Given a feasible point of X_A, a typical cutting plane algorithm generates a cut which deletes this point but no other point of X_A with a quadratic objective value better than the current best. The algorithm proceeds by searching for another point of X_A which is feasible to the cuts generated thus far. If none exists, then the incumbent is declared optimal. Otherwise, a suitable improvement routine may be applied to the new feasible point, the current best solution is updated if necessary, and the above procedure is repeated.

Intersection Cuts

Let $\bar{x} \varepsilon X_A$ be feasible to all previously generated cuts. Consider the extended simplex tableau yielding a basic representation of $\bar{x}$ in terms of the constraints in X and not including any cutting planes. Correspondingly, let J denote the set of nonbasic variables and note that $|J| = (m-1)^2$. Identify the $(m-1)^2$ edges incident at $\bar{x}$, each edge associated with a single nonbasic variable, and let $\bar{a}^j$ be the extended column of the nonbasic variable x_j. It can be shown that the following is a valid inequality:

$$\sum_{j \varepsilon J} x_j/\lambda_j \geq 1$$

where

$$0 < \lambda_j < [-d_{2j} + \sqrt{d_{2j}^2 + 4d_{1j}d_3}\ ]/2d_{1j} \qquad j \varepsilon J$$

$$d_{1j} = -(\bar{a}^j)^t D\bar{a}^j > 0$$

$$d_{2j} = 2(\bar{a}^j)^t D\bar{x}$$

$$d_3 = \bar{x}^t D\bar{x} - \bar{\nu} > 0$$

$\bar{\nu}$ = incumbent objective value -1.

If all nondegenerate pivots lead to no improvement in the objective function value, then the above cut can be strengthened to:

$$\sum_{j \in J} x_j \geq 2$$

Disjunctive Cuts

Given a feasible assignment x, if all pairwise exchanges produce no improvement in the objective value, then the following disjunctive cost is valid:

$$\sum_{i=1}^{m} x_{ia(i)} \leq m-3$$

where a(i) is the location to which object i is assigned.

Another disjunctive cut based on cost considerations is given by:

$$\sum_{i=1}^{m} \sum_{j=1}^{m} u_{ij} x_{ij} \leq \bar{v}$$

where

$$u_{ij} = \underset{\substack{x \in X_A \\ x_{ij} = 1}}{\text{minimum}} \sum_{k=1}^{m} \sum_{\ell=1}^{m} d_{ijk\ell} x_{k\ell}$$

2.2. Number of Cuts

The first disjunctive cut deletes $1 + \frac{m(m-1)}{2}$ points of X_A. It deletes the current point as well as those points in X_A that are generated through pairwise exchanges. Empirically, from computational experience, it is found that for most problems with $m \geq 7$, maximum $\{u^t x: x \epsilon X_A\} < \hat{v}$, where $\hat{v}$ is the best known objective for the quadratic assignment problem. This essentially states that the second disjunctive cut does not delete any points in X_A.

Finally, the intersection cut deletes the current point and those points in X_A which are obtained via a single nondegenerate pivot since any other point in X_A must have at least two of the variables x_j for $j \epsilon J$ equal to one. It can be shown that the maximum number of nondegenerate pivots from any basis representing the current point is $\frac{m(m-1)}{2}$. Hence it follows that the intersection cut deletes at most $1 + \frac{m(m-1)}{2}$ points.

Thus a lower bound on the number of cuts needed for termination is:

$$\frac{m!}{1 + \frac{m(m-1)}{2}} \cong 2(m-2)!$$

2.3. A Heuristic Method

Cutting planes can be utilized to develop an efficient heuristic for solving the quadratic assignment problem. Particularly, suppose that c cuts of the form $\alpha_i^t x \geq \theta_i$ for $i=1,\ldots,c$ have already been generated. A point $x \epsilon X_A$ that satisfies these cuts is sought. If suitable weights $w_1,\ldots,w_c$ are chosen, the fictitious linear assignment problem to

$$\begin{aligned} &\text{Maximize} \quad \sum_{i=1}^{c} w_i \alpha_i^t x \\ &\text{Subject to} \quad x \epsilon X \end{aligned}$$

often produced a solution which is feasible to the cuts. The weight w_i used is given by $1/|\underset{x\varepsilon X}{\text{maximum}}\ \alpha_i^t x\ |$ so that each cut is normalized by dividing it by its maximum absolute value. The fictitious objective function thus seeks a point in X_A which is roughly equidistant from each cut. It is also found helpful to add the disjunctive cost cut into the objective function via a suitable weight w leading to the following problem:

$$\begin{array}{ll} \text{Maximize} & \beta^t x \\ \text{Subject to} & x\varepsilon X_A \\ \text{where} & \beta = \sum_{i=1}^{c} w_i\alpha_i - wu \end{array}$$

A new cut c+1 is generated using the optimal solution to the above problem and the process is repeated.

3. BRANCH AND BOUND METHODS

The main feature of the branch and bound algorithm is the elimination of "mirror image" branches in the search tree. The procedure is modified in order to accelerate the computations resulting in an efficient heuristic procedure with the following characteristics:

1. Several improvement routines are used in conjunction with the branch and bound scheme. The extent of using these improvement routines is a function of the branch and bound tree level.

2. Several heuristics are utilized to eliminate the search effort at branches which are likely not to lead to objective value improvements. Furthermore, variable upper bounds are used to reduce the number of solutions examined.

3.1. An Exact Branch and Bound Procedure

A single assignment branch and bound scheme that uses the lower bounding scheme of Gilmore and Lawler is developed. Since neither the depth nor the breadth branching strategies are satisfactory for the quadratic assignment problem, the proposed algorithm combines both strategies.

The attainment of good quality solutions early on is of great importance, especially if the algorithm is eventually used as a heursitic. The correlation between lower bounds and quality of partial assignments at low levels of the branch and bound tree is not strong. Thus it is highly likely that a depth strategy may select poor quality branches to pursue initially so that good quality solutions are obtained only after a large number of nodes is evaluated. On the other hand, high levels of the branch and bound tree are not reached early on if the breadth strategy is used. Since good quality solutions are usually obtained only at high levels of the tree, the process of obtaining such solutions is also delayed. For this reason, the proposed algorithm combines the two strategies. Particuarly, a breadth strategy is used as long as the tree level L has not reached L_1 for the first time. The depth strategy is used if $L > L_1$. With this combined strategy, many candidate good quality partial solutions are formed at low tree levels. Starting with one of these solutions, the depth strategy quickly finds good quality complete solutions.

3.2. A Branch and Bound Based Heuristic Algorithm

Several improvement routines and methods of eliminating certain branches which are likely not to produce good quality solutions are incorporated in the branch and bound method resulting in a heuristic procedure.

Particularly, in order to improve upper bounds, exchange routines are applied to the solution of the linear assignment problem used to compute a lower bound. Furthermore, since it is not possible to exhaust the search tree for large problems, several heuristics including variable upper bounds are used for discontinuing the search at branches where improvements are not likely even if the lower bounds indicate that fathoming is not yet possible.

APPENDIX

This appendix consists of the following three research manuscripts conducted under this grant:

1. M. S. Bazaraa and H. D. Sherali, On the Use of Exact and Heuristic Cutting Plane Methods for the Quadratic Assignment Problem.

2. M. S. Bazaraa and H. D. Sherali, Cutting Plane Methods for the Quadratic Assignment Problem.

3. M. S. Bazaraa and O. Kirca, A Branch and Bound Based Heuristic for Solving the Quadratic Assignment Problem.

Cutting Plane Methods for the Quadratic Assignment Problem [+]

Mokhtar S. Bazaraa and Hanif D. Sherali
School of Industrial and Systems Engineering
Georgia Institute of Technology

Abstract

This paper uses the formulation of the quadratic assignment problem as that of minimizing a concave quadratic function over the assignment polytope. Cutting plane procedures are proposed for this problem. In particular, efficient methods of generating intersection and disjunctive cuts, deeper than normally available, are discussed. A lower bound derived on the number of cuts needed for termination indicates that cutting plane procedures would require a huge computational effort for the exact solution of quadratic assignment problems. However, through computational testing, the procedures are shown to yield good heuristics capable of detecting optimal or good quality solutions early on in the search process.

[+] This research is supported under NSF grants ENG 77-07468 and ENG 79-08375

1. Introduction

This study addresses the use of cutting plane methods for solving the quadratic assignment problem, which involves the assignment of m indivisible entities, called facilities, to m mutually exclusive locations. The facilities may be plants, warehouses, departments, machines, circuit components, system elements or team individuals, whereas the locations may be physical as in potential plant sites, logical as in natural ordering, or qualitative as in task assignments. The objective is to minimize a quadratic functional which reflects not only the fixed cost of assigning each facility to some location, but also the interaction cost accruing from the location of each facility relative to the location of other facilities.

This problem was first formulated by Koopmans and Beckmann [13]. A mathematical formulation of its generalization due to Graves and Whinston [11] is given below:

$$\text{QAP 1:} \quad \textit{minimize} \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij}\, x_{ij} + \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} \sum_{n=1}^{p} b^n_{ijk\ell}\, x_{ij}\, x_{k\ell}$$

$$+ \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} \sum_{n=1}^{p} f^n_{ik}\, d^n_{j\ell}\, x_{ij}\, x_{k\ell}$$

subject to

$$x \epsilon X_A = \{(x_{11}, \ldots, x_{mm}): \sum_{i=1}^{m} x_{ij} = 1,\ j=1, \ldots, m, \sum_{j=1}^{m} x_{ij} = 1,$$

$$i=1, \ldots, m,\ x_{ij} = 0 \text{ or } 1,\ i,j=1, \ldots, m\}$$

Essentially, this is a multi-commodity problem in which p products flow among m facilities. Accordingly, f^n_{ik} is the amount of flow of product n from facility i to facility k and $d^n_{j\ell}$ is a distance measure from location j to location ℓ when transporting product n. Further, c_{ij} is the fixed cost of locating facility i at location j, and $b^n_{ijk\ell}$ is a fixed cost for product n, dependent on a pair of assignments, viz, facility i to location j and facility k to location ℓ. Note that without loss in generality, we can take $b^n_{ijk\ell} = 0$ if $i=k$ or $j=\ell$ and also, $f^n_{ii} = d^n_{jj} = 0$ for $i,j = 1,\ldots,m$, $n = 1,\ldots,p$.

Using a simple transformation introduced by Lawler [15], and extended by Pierce and Crowston [18], the above problem may be written as

$$\textit{QAP 2: minimize } \{x^t Sx = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} s_{ijk\ell}\, x_{ij}\, x_{k\ell} : x \varepsilon\ X_A\}$$

$$\textit{where, } s_{ijk\ell} = \begin{cases} \sum_{n=1}^{p} (b^n_{ijk\ell} + f^n_{ik}\, d^n_{j\ell}) & \text{if } i \neq k \text{ or } j \neq \ell \\ c_{ij} & \text{otherwise} \end{cases} \tag{1}$$

and where a superscript t will be used throughout to denote the transpose operation. Now, observe that in Problem *QAP 2*, we may assume that S is symmetric for, if not, we may simply achieve this by replacing S by $1/2\ S + 1/2\ S^t$. Further, Problem *QAP 2* may be transformed into a concave quadratic programming problem by simply subtracting a sufficiently large positive constant M from each of the main diagonal elements of S. The equivalence between *QAP 2* and the resulting problem

$$\textit{QAP 3: minimize } \{x^t\ Dx:\ \ x \varepsilon\ X_A\}$$

where $D = S - M\ I$, I being an identity matrix of size m^2, is well known and can be easily established by noting that this operation merely affects the objective function by a constant for any extreme point solution. That is, for any $x \varepsilon\ X_A$,

$$x^t\ Dx = x^t\ Sx - M\ x^t x = x^t\ Sx - M\ m \tag{2}$$

Lemma 1 below establishes that it is sufficient to take a value for M which is greater than the largest sum of elements in any row of S.

Lemma 1

Let $S = (s_{ij})$ be a symmetric, square, non-negative matrix of size n and let $D = S - IM$ where I is an identity matrix of size n and M is a non-negative scalar. Then D is negative definite if

$$M = 1 + \underset{i\varepsilon\{1,\ldots,n\}}{\text{maximum}} \sum_{j=1}^{n} s_{ij} \tag{3}$$

Proof

$$x^t Dx = \sum_{i=1}^{n} d_{ii}\, x_i^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}\, x_i\, x_j = \sum_{i=1}^{n} [d_{ii} + \sum_{\substack{j=1 \\ j\neq i}}^{n} d_{ij}]\, x_i^2$$

$$- \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij}\, (x_i - x_j)^2$$

$$= \sum_{i=1}^{n} [-M + \sum_{j=1}^{n} s_{ij}]\, x_i^2 - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s_{ij}\, (x_i - x_j)^2 \leq \sum_{i=1}^{n} [-M + \sum_{j=1}^{n} s_{ij}] x_i^2$$

Thus, $x^t Dx < 0$ for any $x \neq 0$ for M specified in (3). This completes the proof.

Now, let us denote by X the assignment polytope

$$X = \{x = (x_{11},\ldots,x_{mm}): \sum_{i=1}^{m} x_{ij}=1,\ j=1,\ldots,m,\ \sum_{j=1}^{m} x_{ij}=1,\ i=1,\ldots,m,$$

$$x_{ij} \geq 0,\ i,j=1,\ldots,m\} \tag{4}$$

It is well known that extreme points of X are in a one-to-one correspondence with points in X_A. Further, with M chosen through Lemma 1, since $x^t Dx$ is concave, it is also well known that Problem *QAP 3* is equivalent to

QAP 4: minimize $\{x^t Dx:\ x \varepsilon X\}$

that is, *QAP 4* attains an optimal solution at an extreme point of X. However, locally optimal solutions need not necessarily be globally optimal. We hence propose to employ cutting plane schemes.

Specifically, we investigate the use of two types of cutting planes. The first type of cuts are intersection cuts based on level sets and are a specialization of Tui's [21] cutting planes. The second type of cuts are disjunctive cuts based on suitable logical statements. Although these are derived directly, they fall under the general disjunctive cut principles afforded by Balas [2] and Jeroslow [12] and in a related discussion, by Glover [10]. The use of other types of cutting planes is also discussed and extensive computational results are provided. Before that, let us discuss a general framework in which these cutting plane procedures may be imbedded.

2. General Framework for the Proposed Cutting Plane Procedures

As an expedient in the discussions to follow, we will denote the vector x as

$$(x_{11},\ldots,x_{1m},x_{21},\ldots,x_{m1},\ldots,x_{mm}) = (x_1,\ldots,x_m,x_{m+1},\ldots,x_{m^2})$$

The proposed framework is presented below in algorithmic form. The details of Step 1 are the subject of discussion of the next two sections, whereas Step 2 is discussed in Section 5.

Initialization: Find a good starting extreme point solution to problem *QAP 4*. Let c denote the number of cuts generated thus far. Currently, c=0.

Step 1: Given a point $\bar{x} \in X_A$ feasible to the cuts, generate another valid cut which deletes $\bar{x}$, but no point of X_A with a quadratic objective value better than the current best. Increment c by one.

Step 2: Let Q denote the set of points feasible to the c cuts $\alpha_i^t x \geq \theta_i$, $i=1,\ldots,c$ generated thus far. Find a feasible point $\bar{x}$ in the set $Q \cap X_A$. If none exist, terminate with the current best solution as optimal. Otherwise, return to *Step 1*.

Note that a scheme based on the above framework is finitely convergent since X_A is a finite set and *Step 1* deletes at least one point of X_A per iteration.

3. Intersection Cuts from Level Sets

In this section, we specialize Tui's [21] basic cutting plane scheme to generate cuts not only in a more efficient manner, but also to obtain cuts deeper than usually available.

Let us hence begin by briefly discussing Tui's [21] cutting plane generation scheme with minor modifications. Suppose we have a point $\bar{x}$ of X_A feasible to the cuts generated thus far. Consider an extended simplex tableau yielding a basic representation of $\bar{x}$ in terms of only the constraints in X and not including any cutting planes. Correspondingly, let J denote the set of nonbasic variables and note that $|J| = (m-1)^2$. Identify the $(m-1)^2$ edges incident at $\bar{x}$, each edge associated with a single nonbasic variable, and let $\bar{a}^j$ be the extended column of the nonbasic variable x_j, $j\epsilon J$, in the tableau under consideration. Hence, we may write any $x\epsilon X$ as

$$x = \bar{x} - \sum_{j\epsilon J} \bar{a}^j \lambda_j , \qquad \lambda_j \geq 0 \text{ for each } j\epsilon J \tag{5}$$

Further, define a halfline corresponding to each edge incident at $\bar{x}$ according to

$$\xi^j = \{x: x = \bar{x} - \bar{a}^j \lambda_j,\ \lambda_j \geq 0\} \text{ for each } j\epsilon J \tag{6}$$

Now, let ν be the current best objective function value for Problem *QAP 4* and let $\bar{\nu} = \nu - 1$. Consider the level set

$$L(\bar{\nu}) = \{x: \ x^t Dx \geq \bar{\nu}\} \tag{7}$$

Assuming that the matrix D consists entirely of integers, note that $L(\bar{\nu})$ is a convex set which contains $\bar{x}$ in its interior and which does not contain in its interior any point of X_A which has an objective function value better than ν, that is, a value lesser than or equal to $\bar{\nu}$. Hence a valid intersection cut may be derived from $L(\bar{\nu})$ as in [1] or [9]. This cut is defined by a hyperplane passing through the $(m-1)^2$ distinct points of intersection of the halflines ξ^j, $j\varepsilon J$ with $L(\bar{\nu})$ and is given by

$$\sum_{j\varepsilon J} x_j/\bar{\lambda}_j \geq 1 \tag{8}$$

where,

$$\bar{\lambda}_j = \text{supremum } \{\lambda_j: \bar{x} - \bar{a}^j \lambda_j \subset L(\bar{\nu})\}, \text{ for each } j\varepsilon J \tag{9}$$

We next demonstrate that $\bar{\lambda}_j$ is a finite, positive scalar for each $j\varepsilon J$. Note that, for each $j\varepsilon J$, $\bar{\lambda}_j$ is obtained as a solution λ_j to the equation

$$(\bar{x} - \lambda_j\bar{a}^j)^t \, D \, (\bar{x} - \lambda_j\bar{a}^j) = \bar{\nu}$$

or
$$d_{1j}\lambda_j^2 + d_{2j}\lambda_j - d_3 = 0 \tag{10}$$

where,
$$d_{1j} = -(\bar{a}^j)^t \, D \, \bar{a}^j > 0, \quad d_{2j} = 2(\bar{a}^j)^t \, D \, \bar{x}, \text{ and}$$
$$d_3 = \bar{x}^t \, D \, \bar{x} - \bar{\nu} > 0 \text{ for each } j\varepsilon J \tag{11}$$

Thus, $\bar{\lambda}_j$ is the positive root of (10) or,

$$0 < \bar{\lambda}_j = (-d_{2j} + \sqrt{d_{2j}^2 + 4d_{1j}d_3}\,)\,/2d_{1j} < \infty, \text{ for each } j\varepsilon J \tag{12}$$

We remark that the cut given through Equations (8) and (12) is valid even in the presence of degeneracy (see Balas [1]). In fact, the order of degeneracy is (m-1) for each extreme point of X, and this raises another question relating to the choice of a basis to represent a given extreme point $\bar{x}$ of X. As shown by Balinski and Russakoff [5], there are $2^{m-1}m^{m-2}$ possible choices. We now address the question of selecting a suitable basis.

Observe that each choice of a basis representing $\bar{x}$ defines a cone which spans the set X and which has its vertex at $\bar{x}$ and has edges corresponding to the nonbasic variables. Now, the tighter the cone which spans the set X, the deeper the cuts one may expect to derive therefrom. This philosophy is illustrated in Figure 1 below. Here a basis corresponding to edges 2 and 3 yields a cut significantly deeper than a basis corresponding to edges 1 and 3, where edge 1 corresponds to a degenerate pivot.

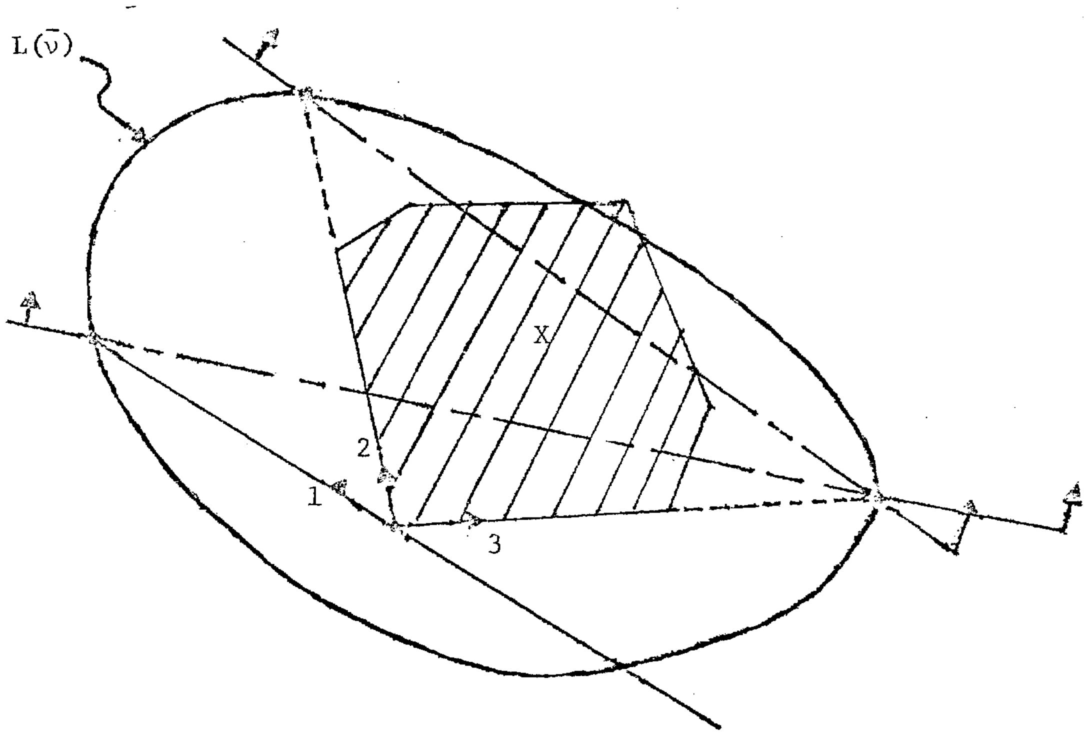

Figure 1 Effect of the choice of a basis on the depth of cut

It stands to reason then, that a desirable basis is one which among all alternate bases, yields the maximum number of non-degenerate pivots. Indeed, we are able to show, through Theorem 1 below, that the maximum number of non-degenerate pivots associated with any feasible basis is $\frac{m(m-1)}{2}$, and this maximum is attained for feasible bases which have chain graphs as their spanning tree representations. Figure 2 depicts one such basis with a spanning tree $T(\bar{x})$, representing a solution $\bar{x} \varepsilon X_A$. Here, $a(i)$ denotes the location of facility i, that is, $\bar{x}_{i,a(i)} = 1$ for $i=1,\ldots,m$

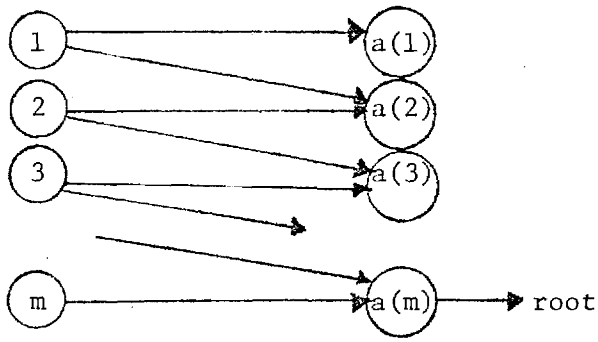

Figure 2 Rooted spanning tree $T(\bar{x})$

Theorem 1 Let $\bar{x}$ be an extreme point of X with $\bar{x}_{i,a(i)} = 1$, for $i=1,\ldots,m$, and let $T(\bar{x})$ be a rooted spanning tree with basic arcs $(i, a(i))$, $i=1,\ldots,m$ and $(i, a(i+1))$, $i=1,\ldots, m-1$. (See Figure 2) Then,

(a) $T(\bar{x})$ results in precisely $\frac{m(m-1)}{2}$ non-degenerate pivots.

(b) Of all alternate bases representing $\bar{x}$, the maximum number of non-degenerate pivots from any such basis is $\frac{m(m-1)}{2}$.

Proof:

(a) It is clear from $T(\bar{x})$ that the only nonbasic arcs leading to non-degenerate pivots are of the form $(i, a(t))$ where $1 \leq t \leq i-1$ for each $i = 2,\ldots,m$. Thus, $T(\bar{x})$ has $\sum_{i=2}^{m} (i-1) = \frac{m(m-1)}{2}$ nondegenerate pivots associated with it. This proves part (a).

(b) We will establish this result by induction. The result clearly holds for m=2. Hence, suppose its true for (m-1) facilities. We will show that this implies that the result is true for an $\bar{x} \varepsilon X_A$ involving m facilities.

Thus, consider any rooted spanning tree representing $\bar{x}$. To prove the result, we will first need to show that this tree contains at least one pair of nodes $\{k, a(k)\}$, $k\varepsilon\{1,\ldots,m\}$ for which the degree of one of the nodes in this pair is one and that of the other is two. (By degree of a node, we mean the number of arcs incident at that node).

Observe that since $\bar{x}_{i,a(i)} = 1$, $i=1,\ldots,m$, and since a tree is a connected graph, each pair of nodes $\{i, a(i)\}$, $i=1,\ldots,m$ has a total degree of at least three. By contradiction, if no pair of the above type exists, then each pair of nodes $\{i, a(i)\}$, $i=1,\ldots,m$ must have a total degree of at least four, and so, the sum of the degrees of all nodes in the tree must be at least $4m$. But any spanning tree has $2m-1$ arcs which implies a total degree of $4m-2$, a contradiction.

Hence, a pair of nodes $\{k,a(k)\}$ exists with say, without loss of generality, degree equal to one for node k and equal to two for node $a(k)$ and with $x_{p,a(k)}$, $p \neq k$, basic at value zero. Now, consider non-degenerate cycles involving the basic arc $(k,a(k))$. Such cycles must involve this arc in the reverse direction and must also be associated with a nonbasic variable arc $(k, a(q))$ for some $q \neq k$. Thus, there are at most $(m-1)$ such non-degenerate cycles. Now, let us compute the maximum number of non-degenerate cycles which do not involve arc $(k, a(k))$.

For this purpose, remove the arcs $(k,a(k))$, $(p,a(k))$ and the nodes k, $a(k)$ from the given graph. The resulting graph has $2(m-1)$ nodes, $2(m-1)-1$ arcs and contains no cycles, that is, it is a spanning tree on the $2(m-1)$ nodes. Moreover, it represents an assignment solution $\bar{x}_{i,a(i)} = 1$, $i\varepsilon\{1,\ldots,m\}$, $i \neq k$. Hence, by our hypothesis, the maximum number of non-degenerate cycles not involving arc $(k,a(k))$ are $\frac{(m-1)(m-2)}{2}$. Therefore, the original tree has at most

$$\frac{(m-1)(m-2)}{2} + (m-1) = \frac{m(m-1)}{2}$$

nondegenerate pivots associated with it. This completes the proof.

It is easy to see that for a given $\bar{x} \varepsilon X_A$, there are m! spanning trees, of the total $2^{m-1}\ m^{m-2}$ possible spanning trees, which are chain graphs. The question as to which of these should be selected is not obvious. Since they all seem to be equally attractive, let us henceforth work with the graph $T(\bar{x})$ of Figure 2.

Some Computational Expedients in Deriving Cuts from $T(\bar{x})$

Observe from Figure 2 that the nonbasic variable arcs $j \varepsilon J$ may be classified into two cases.

Case (i) Degenerate Pivots: $j \varepsilon J_d \subset J$ corresponds to a nonbasic arc $(i, a(t))$, $i+2 \leq t \leq m$ for some $i \varepsilon \{1, \ldots, m-2\}$. Thus,

$$\bar{a}^j \text{ has } \left\{ \begin{array}{l} +1 \text{ for arcs } (i,a(i+1)),\ (i+1,a(i+2)),\ \ldots,\ (t-1,\ a(t)) \\ -1 \text{ for arcs } (i+1,\ a(i+1)),\ (i+2,\ a(i+2)),\ \ldots,\ (t-1,a(t-1)) \text{ and for} \\ \quad \text{arc } (i,\ a(t)) \\ 0 \quad \text{otherwise} \end{array} \right\} \tag{13}$$

Case (ii) Non-degenerate Pivots: $j \varepsilon J_{nd} = J - J_d$, corresponds to a nonbasic arc $(i, a(t))$, $1 \leq t \leq i-1$ for some $i \varepsilon \{2, \ldots, m\}$. Thus,

$$\bar{a}^j \text{ has } \left\{ \begin{array}{l} +1 \text{ for arcs } (t,\ a(t)),\ (t+1,\ a(t+1)),\ \ldots,\ (i,\ a(i)) \\ -1 \text{ for arcs } (t,\ a(t+1)),\ (t+1,\ a(t+2)),\ \ldots,\ (i-1,\ a(i)) \text{ and for} \\ \quad \text{arc } (i,\ a(t)) \\ 0 \quad \text{otherwise} \end{array} \right\} \tag{14}$$

From Equations (13) and (14), it is apparent that it is not necessary to compute d_{1j} and d_{2j} of Equation (11) separately for each $j \varepsilon J$, in that, certain recursive forms are available to derive these values one from another. We avoid writing out explicitly these recursive forms, since they are notationally cumbersome, but emphasize that they are computationally easy to implement and conserve a good deal of effort. However, we illustrate these relationships through an example.

Consider the updated column of Equation (13) for, say, the nonbasic arc $(1,\ a(3))$. This has a (+1) for arcs $(1,\ a(2))$ and $(2,\ a(3))$ and a (-1) for arcs $(2, a(2))$ and $(1,\ a(3))$. Now consider the updated column for the nonbasic arc (1, a(4)). This is identical to that for (1, a(3)) except that the arc (1, a(4)), instead of (1, a(3)), has a (-1) and two new arcs have non-zero values associated with them. Namely, (3, a(4)) has a (+1) and (3, a(3)) has a (-1). Thus, when computing d_{1j} and d_{2j} for the nonbasic variable (1, a(4)), we simply need to accordingly modify the values of d_{1j} and d_{2j} obtained for the nonbasic variable (1, a(3)). Summarizing, in Case (i) above, for each $i=1,\ldots,m-2$, the values of d_{1j} and d_{2j} may be obtained recursively for the nonbasic arcs (i, a(t)), $t = i+2,\ldots,m$. Similarly, in Case (ii) above, for each $t = 1,\ldots,m-1$, the values of d_{1j} and d_{2j} may be obtained recursively for the nonbasic arcs (i, a(t)), $i = t+1,\ldots,m$.

On Validly Deepening the Basic Cut

We will now modify the cut given through Equations (8) and (12) to make it as deep as possible. First of all, note that for $j \varepsilon J_{nd}$ of Case (ii), a value $\lambda_j = 1$ in Equation (6) corresponds to the extreme point of X adjacent to $\bar{x}$ which is reached through the non-degenerate pivot on entering x_j into the basis. Thus, if we find that $\bar{\lambda}_j \leq 1$ for any $j \varepsilon J_{nd}$, then we have detected an improved solution and we abort the cut generation at the current point and move to this new point, which must lie in $X_A \cap Q$ by the definition of a valid cut. Hence, when a cut is finally generated, we must have

$$1/\bar{\lambda}_j < 1 \text{ for each } j \varepsilon J_{nd} \tag{15}$$

Now, consider zero-one solutions in relation to the cuts

$$\sum_{j \varepsilon J_d} x_j/\bar{\lambda}_j + \sum_{j \varepsilon J_{nd}} x_j/\bar{\lambda}_j \geq 1 \tag{16}$$

and

$$\sum_{j \varepsilon J_d} 2x_j + \sum_{j \varepsilon J_{nd}} x_j \geq 2 \tag{17}$$

We will now validate the cut (17). In order to do this, it is sufficient to show that any zero-one solution deleted by (17) is also infeasible to (16).

Particularly, if (17) is violated, then $x_j = 0$ for each $j\varepsilon J_d$ and $\sum_{j\varepsilon J_{nd}} x_j \leq 1$. This further implies that either $\sum_{j\varepsilon J_{nd}} x_j = 0$ in which case (16) is violated or else $\sum_{j\varepsilon J_{nd}} x_j = 1$. In this latter case, and noting from (15) that $1/\bar{\lambda}_j < 1$ for each $j\varepsilon J_{nd}$, it follows that $\sum_{j\varepsilon J_{nd}} x_j/\bar{\lambda}_j < 1$, again violating (16). This shows that the cut given by (17) is indeed valid.

Now, due to the effect of the magnitude of M given by Equation (3), we found through computational testing that invariably we obtain

$$\begin{aligned} 1/\bar{\lambda}_j &> 1 \qquad \text{for } j\varepsilon J_d \\ 1/2 < 1/\bar{\lambda}_j &< 1 \qquad \text{for } j\varepsilon J_{nd} \end{aligned} \tag{18}$$

In fact, for $j\varepsilon J_{nd}$, $1/\bar{\lambda}_j$ was typically greater than 0.9 for $m \geq 12$. This empirical observation supports the use of the cut (17) which seems to be uniformly deeper than the cut (16). Further, it is numerically well conditioned. More importantly, it avoids completely the computations of $\bar{\lambda}_j$ for $j\varepsilon J_d$ and even for $j\varepsilon J_{nd}$, one simply needs to compare the value of $1/\bar{\lambda}_j$ with unity through the following relationship derived from Equation (12)

$$1/\bar{\lambda}_j < 1 \text{ if and only if } d_{1j} + d_{2j} < d_3,\ j\varepsilon J_{nd} \tag{19}$$

It turns out that the cut (17) itself may be further strengthened. Observe that zero-one points feasible to (17) which have exactly one variable equal to unity do not correspond to extreme points of X since they correspond to degenerate pivots. Hence, extreme points of X feasible to (17) must have at least two nonbasic variables equal to unity. In other words, we may let

$$\alpha_i^t x \geq \theta_i \quad \equiv \quad \sum_{j \in J} x_j \geq 2 \tag{20}$$

be the desired cut.

To summarize the cut generation scheme, one simply needs to verify (15) by using (19). If (15) holds, then the cut (20) may be derived. Otherwise, an improved feasible solution is detected and the cut generation scheme is reactivated at this new point.

4. Disjunctive Cutting Planes

In this section, we will use the methods of disjunctive programming to derive valid inequalities for our problem. A linear disjunction is a logical statement which asserts that at least p of q linear inequality systems must be satisfied. We will state two such disjunctions and derive appropriate cuts therefrom. The material which follows is self-contained, and a reader interested in disjunctive programming principles is referred to [2,3,10,12].

To begin with, consider an extreme point $\bar{x}$ of X feasible to the cuts generated thus far and let $\bar{x}_{i,a(i)} = 1$ for all i. Note that any other extreme point of X has at least two of the variables $x_{i,a(i)}$, for $i=1,\ldots,m$, equal to zero. Further, suppose that $\bar{x}$ is such that any pairwise interchange on it results in either a non-improving solution or a solution infeasible to the cuts generated thus far. We may hence assert the following disjunction:

<u>DC 1</u> At least three of the variables $x_{i,a(i)}$ must be equal to zero. That is, at least 3 of the following m inequality systems must be satisfied

$$\{x_{i,a(i)} \leq 0, \quad x \geq 0\}, \quad i = 1,\ldots,m$$

A strongest, valid disjunctive cut based on DC1, that is, one which deletes all points not satisfying DC1, but none satisfying this statement, is easily obtained as

$$\sum_{i=1}^{m} x_{i,a(i)} \leq m-3 \tag{21}$$

We will now state a second disjunction, DC2, based on the objective function of Problem *QAP 4* as opposed to DC1 which is based on the constraints of this problem. Towards this end, consider the reverse polaroid

$$X^0(\bar{\nu}) = \{x: x^t D x^r \geq \bar{\nu} \text{ for each } x^r \varepsilon X_A\} \tag{22}$$

where $\bar{\nu} = \nu - 1$, and ν is the known current best objective function value of Problem *QAP 4*. Clearly, the interior of $X^0(\bar{\nu})$ contains no extreme point of X with an objective function value less than $\bar{\nu}$. We may hence assert the following disjunction:

DC2 At least one of the following m! linear inequality systems must be satisfied

$$\{\sum_{i=1}^{m} \sum_{j=1}^{m} (\sum_{k=1}^{m} \sum_{\ell=1}^{m} d_{ijk\ell}\, x^r_{k\ell})\ x_{ij} \leq \bar{\nu},\ x \geq 0\},\ r=1,\ldots,m!,\ x^r \varepsilon X_A$$

Hence, for DC2 to be satisfied, since $x \geq 0$, we must have,

$$\hat{u}^t x \equiv \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\underset{x^r \varepsilon X_A}{\text{minimum}} \sum_{k=1} \sum_{\ell=1} d_{ijk\ell}\, x^r_{k\ell}\right] x_{ij} \leq \bar{\nu} \tag{23}$$

It is easy to see from the definition of D and from Equations (2) and (3) that for points in X_A, the cut (23) is identical to the cut

$$u^t x \equiv \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\underset{\substack{x^r \varepsilon X_A \\ x^r_{ij}=1}}{\text{minimum}} \sum_{k=1}^{m} \sum_{\ell=1}^{m} s_{ijk\ell}\, x^r_{k\ell}\right] x_{ij} \leq (\bar{\nu} + mM) = \bar{\bar{\nu}}, \text{ say} \tag{24}$$

In other words, $u^t x \leq \bar{\bar{\nu}}$ is a valid inequality for DC 2. It is interesting to note here that $u^t x$ is precisely the linear bound derived by Cabot and Francis [6] for the quadratic assignment problem. Some further comments on this bounding rule are afforded in the following two sections. Observe that the cut (23) depends only on the current best objective function value and is not derived from any particular point in X_A.

To summarize, the cut (21) is generated each time a feasible point $\bar{x} \varepsilon X_A$ is located such that pairwise interchanges on $\bar{x}$ lead to non-improving or infeasible points. Further, the cut (24) has its right-hand-side updated each time an improved solution is detected.

5. Exact Cutting Plane Solution Procedures

In this section, we first address the execution of Step 2 in the framework of Section 2. Thereafter, we critically examine the cuts (20), (21) and (24).

Hence, let

$$Q = \{x\colon \alpha_i^t x \geq \theta_i,\ i=1,\ldots,c\}$$

be the set of points feasible to the c cuts generated thus far. Further, let

$$\bar{\alpha}_i = \text{maximum}\ \{1,\ |\text{maximum}\ \{\alpha_i^t x : x \varepsilon X_A\}|\} \tag{26}$$

and set

$$\beta = \sum_{i=1}^{c} (1/\bar{\alpha}_i)\alpha_i \tag{27}$$

Consider the problem

$$\text{FEAS:}\quad \text{maximize}\ \{\beta^t x\colon\ \ x \varepsilon\ X_A\}$$

The motivation for the above steps is as follows. Problem FEAS maximizes a fictitious objective function formed through (27) by normalizing each cut expression by its maximum value and adding up the resulting expressions. Intuitively, it hence seeks a point in X_A which is roughly equidistant from each cut. In fact, the solution to Problem FEAS is often feasible to Q. This is particularly true if there are several feasible solutions in $X_A \cap Q$ which are yet unexplored.

On the other hand, if the solution to FEAS does not lie in Q, then we may use an implicit enumeration scheme to find a point in $X_A \cap Q$ or conclude that none exist. One may note, that such an implicit enumeration scheme may be initialized just once and then simply updated each time Problem FEAS does not directly locate a feasible point. However, we avoid giving details of such a scheme in view of the following conclusions.

Note that both the cuts (20) and (21) delete precisely $1 + \frac{m(m-1)}{2}$ points of X_A. The cut (20) deletes the point $\bar{x}$ from which it is generated and only those $\frac{m(m-1)}{2}$ points in X_A which correspond to non-degenerate pivots on $T(\bar{x})$.

Similarly, the cut (21) deletes the point $\bar{x}$ from which it is generated and only those $\frac{m(m-1)}{2}$ points in X_A reached through pairwise interchanges on $\bar{x}$. In other words, a lower bound on the number of either types of cuts needed for termination is:

$$\# \text{ of cuts} \geq \frac{m!}{1 + \frac{m(m-1)}{2}} \approx 2(m-2)! \tag{28}$$

For example, for m=7 at least 240 cuts would be required and for m=8, at least 1440 cuts would be required.

Now consider the cut (24). Computationally, we found that for most problems with $m \geq 7$,

$$\underset{x \in X_A}{\text{maximum}}\ u^t x < \hat{\nu} \tag{29}$$

where $\hat{\nu}$ is the best known objective value of *QAP 2*. This essentially states that the cut (24) does not delete any point of X_A for even the smallest valid value of $\overset{=}{\nu}$. Computational evidence using test problems in the literature are given in the next section.

In view of the above considerations, we found it futile to use cutting plane procedures to solve quadratic assignment problems exactly. We merely remark here that we attempted other cutting planes like that derived from reverse outer polar sets [4], another derived through a linear bounding scheme available from Roucairol's reduction method [19] and we also attempted a direct application of Tui's [21] procedure, all in vain. Assuming that the magnitude of M in the objective of *QAP 3* was having an adverse effect on the depth of the cuts, we even devised a technique to find out the largest integer q for which the cut $\sum_{j \in J} x_j \geq q$ is valid for Problem *QAP 2*. Empirically, we were never able to improve on the cut (20).

However, from the methods discussed in the foregoing sections, we were able to obtain heuristics capable of finding optimal or good quality solutions early on in the search process. The following section provides details for these heuristics.

6. Heuristics Derived from Cutting Plane Procedures

The fundamental utility of the heuristic procedures we present in this section is based on two facts. Firstly, let us consider the cut (24) and define

$$\nu_{min} = \text{minimum}\ \{u^t x : x \varepsilon X_A\} \text{ and } \nu_{max} = \text{maximum}\ \{u^t x : x \varepsilon X_A\} \qquad (30)$$

Computationally, we found that if x^* is the best known solution to the quadratic assignment problem, then $\nu^* = u^t x^*$ occurs in the earlier portion of the range $[\nu_{min}, \nu_{max}]$. In fact, we found that ν^* was often contained within 15% of the range $[\nu_{min}, \nu_{max}]$. Table 1 presents the characteristics of ν^* for some well-known test problems.

Table 1. Some Characteristics of the Disjunctive Cost Cut (24)

Problem	m	ν_{min}	ν_{max}	$x^{*t} Sx^*$	$\nu^* = u^t x^*$	$\frac{\nu^* - \nu_{min}}{\nu_{max} - \nu_{min}}$
	5	50	55	50	50	0.0
Nugent, Vollmann and	6	82	84	86	82	0.0
Ruml's Problems [16]	7	137	144	148	140	0.43
	8	186	199	214	188	0.15
	12	493	517	578	496	0.13
	15	963	1,034	1,150	970	0.09
	20	2,057	2,243	2,570	2,070	0.07
	30	4,539	4,948	6,154	4,587	0.12
Steinberg's Problems [20]: Squared Euclidean Distance	36	8,653	14,305	15,852	8,899	0.04
Rectilinear Distance	36	7,124	8,590	9,604	7,198	0.05
Elshafei's Problem[8]	19	11,971,949	25,541,722	17,212,548	12,568,503	0.04
Krarup's Problem [14]	32	67,390	77,680	90,220	69,140	0.17

A second encouraging fact was that Problem FEAS often determined solutions feasible to the cuts or at least close to feasible solutions. We hence exploited the above two observations in the following manner.

In formulating the fictitious objective function of Problem FEAS, we replaced β of Equation (27) by the following expression:

$$\beta = \sum_{i=1}^{c} (1/\bar{\alpha}_i)\alpha_i - \left(\frac{w}{\nu_{min}}\right) u \tag{31}$$

where ν_{min} is given through Equation (30), and w is a suitable parameter which determines the relative weight given to the disjunctive cost cut over the other cuts. In other words, with β given by Equation (31), Problem FEAS attempts to seek out good quality feasible solutions, and in our experience, led to an early detection of desirable solutions.

Figure 3 gives a flow-chart for the proposed heuristic schemes using either cuts (20) or (21). Both these procedures incorporate (31). Further, the cuts are used only to update β and need not be stored. Also an improvement routine using pairwise exchanges on the solution of FEAS is incorporated. We did not attempt three-way interchanges, since as reported by Burkard and Stratmann [6], this is computationally wasteful as compared to the benefits accruing from it. If the pairwise exchange scheme leads to an improved solution, the latter is used to generate the next cut. Otherwise, the next cut is generated from the solution to FEAS. When the disjunctive cut (21) is used, one needs to verify that no single pairwise exchange improves the solution over the current best value at a point from which the cut is being generated. In case of the intersection cut (20), if $1/\bar{\lambda}_j \geq 1$ for $j \epsilon J_{nd}$, then the corresponding improved solution found is adopted as discussed in Section 3.

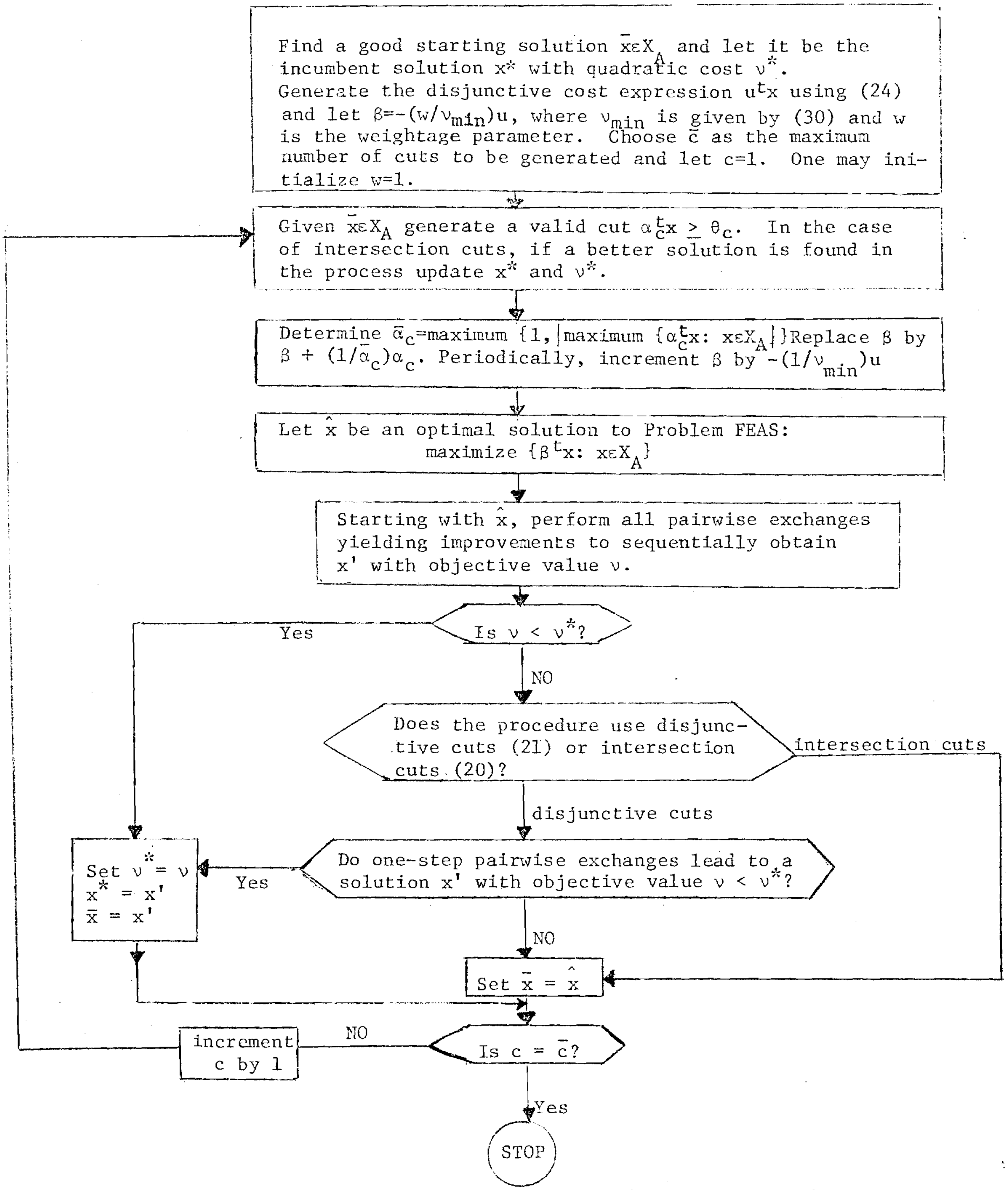

FIGURE 3. Flow Chart for Heuristic Procedures Employing Intersection Cuts or Disjunctive Cuts

Finally, we comment that the performance of this heuristic is sensitive to both the starting solution as well as to the weightage of the parameter w in Equation (31). An initial solution may be obtained by arranging facilities in nonincreasing order of their flow sums and arranging locations in non-decreasing order of their distance sums, and then matching these two arrangements. Pairwise exchanges may be performed on this solution for further improvement. As far as the parameter w in Equation (31) is concerned, we found it usually suitable to initialize with w=1 and then to increment w by one every three to ten cuts. For larger problems, we further recommend a few short trial runs with different rates of incrementing w and different starting solutions as found over previous runs, and then a final run with the best found solution as the starting solution.

7. Computational Experience

In this section, we report our computational experience using both, test problems available in the literature, as well as some randomly generated problems having a special structure such that an optimal solution is known.

Table 2 presents our results with test problems taken from references [8, 14, 16, 20]. For both the cutting plane heuristics, we employed the starting solution obtained by matching ordered flow and distance vectors as described in the foregoing section, and then performing pairwise interchanges on the resulting solution. Column 'a' gives the starting solution values, column 'b' gives the best values obtained on using this initial solution and terminating the procedure after 50 cuts. Column 'c' gives the cut indices at which these best solutions were found. Column 'd' gives the execution time in cpu seconds for a run which generates 50 cuts on a CDC Cyber 70 Model 74-28/CDC 6400 computer, with coding in FORTRAN IV. These times do not include the effort for generating either the starting solution or the expression $u^t x$ of Equation (24).

Through a few subsequent runs employing better quality starting solutions as obtained over previous runs of either procedures, we were able to use these heuristics to further improve upon the initial run solutions. Column 'e' gives the best objective values we were able to obtain in this manner. The quality of these solutions may be compared with the previously best known solution values reported in the literature [6,8] as given in column 'f'. Finally, column 'g' gives the locations of facilities $1,\ldots,m$ respectively, corresponding to the solutions with objective values given in column 'e'.

Before proceeding, we note that better quality solutions may be found on the initial run itself if some more sophisticated exchange schemes such as those proposed by Obata and Mirchandani [17] are used in lieu of the simple pairwise exchange operations that we have been employing.

To further test these heuristic procedures, we generated random problems with known optimal solutions in the following manner. We assumed that the problem data was comprised of simply a flow and a distance matrix $F = (f_{ik})$ and $D = (d_{j\ell})$, respectively. The matrix D was assumed to correspond to a rectangular array of locations, with a rectilinear distance measure being used.

Next, the flow matrix F was constructed in a manner such that the solution $x_{ii} = 1$, $i=1,\ldots,m$ was optimal. To see how this may be achieved, let the location array be $p \times q$ in dimension. Then the matrix D has entries ranging from 1 to $p + q - 2$. Accordingly, define sets $S_1, S_2, \ldots, S_{p+q-2}$ as

$$S_r = \{\text{index pairs } (j,\ell):\ d_{j\ell} = r\} \text{ for } r = 1,\ldots,p+q-2 \qquad (32)$$

Then, select entries for the flow matrix $F = (f_{ik})$ according to

$$(p+q-r-2)\,V + 1 \leq f_{ik} \leq (p+q-r-1)\,V$$

$$\text{for each pair of facilities } (i,k) \in S_r,\ r=1,\ldots,p+q-2 \qquad (33)$$

where V is a prechosen parameter which determines the variance of the flow matrix.

Table 2. Computational Results for the Cutting Plane Heuristics

Problem		m	a	Intersection Cuts (20) b	Intersection Cuts (20) c	Intersection Cuts (20) d	Disjunctive Cuts (21) b	Disjunctive Cuts (21) c	Disjunctive Cuts (21) d	e	f	g
Nugent, Vollmann and Ruml's Problems [16]		5	26	25	2	1.33	25	4	0.60	25	25	3,4,5,1,2
		6	43	43	1	2.61	43	1	1.00	43	43	3,2,1,6,5,4
		7	78	74	3	4.49	74	2	1.03	74	74	5,6,7,1,2,3,4
		8	118	107	4	7.13	107	3	4.16	107	107	3,4,8,2,1,5,6,7
		12	309	289	3	12.9	289	7	10.92	289	289	5,1,9,8,4,3,11,7,10,2,6,12
		15	610	575	6	25.65	575	6	15.30	575	575	15,14,9,10,2,3,7,12,11,5,6,1,13,8,4
		20	1,334	1,290	8	70.88	1,285	29	34.91	1,285	1,287	17,9,2,10,19,16,18,12,1,3,7,8,11,4,14,6, 20,5,15,13
		30	3,185	3,089	16	261.67	3,078	10	198.39	3,077	3,079	11,2,20,26,1,3,16,22,8,10,29,4,9,25,30, 21,12,23,15,19,7,17,24,5,18,6,28,13,14, 27
Steinberg's Problems [20]	Squared Euclidean Distances	36	8,467	7,946	15	542.83	8,291	8	580.01	7,926	7,926	4,19,29,21,30,31,13,20,2,12,32,23,22,24, 3,1,10,11,15,14,34,35,25,36,26,27,33,5, 6,7,8,16,18,17,9,28
	Recti-linear Dis-tances	36	5,213	5,019	24	422.00	4,914	44	514.74	4,802	4,802	23,18,8,16,7,6,24,17,34,25,5,14,15,4,33, 35,27,26,21,13,11,2,12,1,10,19,3,22,32, 31,30,20,28,29,36,9
Elshafei's Problem [8]		19	c_1	c_2	5	73.36	c_2	2	37.90	c_2	c_3	17,18,19,11,12,9,3,14,1,2,10,13,7,5,15, 16,8,4,6
Krarup's Problem [14]		32	47,360	45,485	41	354.11	45,490	16	258.46	45,110	45,210	31,26,25,22,23,21,7,11,12,3,24,20,8,16, 18,27,28,32,1,10,6,19,2,9,14,30,15,4,5, 29,13,17

a, b, c, d, e, f, g, See text of Section 7 for connotation.

$c_1 = 11299400$

$c_2 = 8606274$

$c_3 = 11281888$

In other words, for index pairs (i,k) for which the distance $d_{ik} = p+q-2$ the corresponding flow values f_{ik} were uniformly generated over the range $[1,V]$. Similarly, for index pairs (i,k) for which $d_{ik} = (p+q-2) - 1$, the flows f_{ik} were generated uniformly over $[V+1,2V]$, and so on till finally, for index pairs (i,k) for which $d_{ik} =1$, the flows f_{ik} were uniformly generated over $[(p+q-3)V +1, (p+q-2)V]$. It is easy to see that for this data, the solution $x_{ii} = 1$, $i=1,\ldots,m$ is optimal since it has the same objective value as the lower bound obtained by taking the inner product of the two vectors obtained by respectively arranging the flows and the distances in nonincreasing and nondecreasing order.

Table 3 gives statistics for generating the data as well as the disjunctive cost cut (24). We note that for the latter cut, the coefficients u_{ij} were obtained by solving only approximately the corresponding linear assignment problems defined by (24). This approximate solution was taken to be the sum of the minimal elements in each row and the minimal elements in each column after each row was reduced by its minimal element. Three values of the variance parameter V were attempted.

Table 4 shows the implementation of the cutting plane heuristics on the problems of Table 3. The results obtained were indeed encouraging. The procedures always detected an optimal solution on the very first run except for the single problem of size m = 60 for which only five intersection cuts were developed. Further, the variance parameter V did not seem to influence the performance of these heuristics. We also note, that for these randomly generated problems, an arbitrary initial solution was used, since the procedure of Section 6 for finding the starting solution is biased in favor of the particular manner in which the data was generated. This initial solution placed the facilities $1,\ldots,m$ column by column on the rectangular location grid.

Table 3 Computational Statistics for Generating Test Problems and Disjunctive Cost Cuts

Problem Size m	Generation of Problem Data					Generation of Disjunctive Cost Cut		
	p	q	V	optimal value	time for problem generation*	ν_{min}^{+}	ν_{max}^{+}	time for cut generation*
20	4	5	20	49,862	0.128	65,697	72,567	8.27
30	5	6		182,064	0.234	223,858	246,328	42.34
40	5	8		408,681	0.414	481,555	534,625	138.16
50	5	10		872,583	0.656	980,628	1,103,978	334.41
60	6	10		1,465,814	0.964	1,614,606	1,808,510	706.27
70	7	10		2,293,236	1.286	–	–	958.6**
20	4	5	5	10,613	0.127	13,728	15,244	10.38
30	5	6		38,925	0.279	47,132	52,098	41.94
40	5	8		100,895	0.412	117,840	131,198	137.39
20	4	5	60	130,504	0.122	172,153	189,997	8.83
30	5	6		476,110	0.246	586,022	644,726	40.78
40	5	8		1,229,567	0.403	1,451,112	1,609,808	129.37

+ ν_{min}, ν_{max} ≡ respectively the minimum and maximum vlaue of the disjunctive cost cut expression

* seconds on a CDC Cyber 74 computer.

** Terminated prematurely.

Table 4 Computational Experience with the Proposed Heuristics Applied to the Randomly Generated Test Problems

Problem Size* m	Optimal Value	Initial Value	Intersection Cuts from Level Sets				Disjunctive Cuts			
			a	b	c	d	a	b	c	d
20	49,862	56,111	10	49,862	4	22.05	10	49,862	7	10.23
30	182,064	201,433	10	182,064	9	127.09	10	182,064	4	59.65
40	408,681	453,935	10	408,681	3	351.15	10	408,681	2	272.34
50	872,583	958,908	10	872,583	2	1168.68	10	872,583	9	521.49
60	1,465,814	1,621,161	5	1,469,773**	3	1012.16	10	1,465,814	2	1076.27
70	2,293,236	2,513,666	-	-	-	-	5	2,293,236	2	958.89
20	10,613	11,967	10	10,613	9	21.66	10	10,613	3	10.30
30	38,925	43,160	10	38,925	6	149.83	10	38,925	4	57.83
40	100,895	112,214	10	100,895	4	425.38	10	100,895	6	227.83
20	130,504	146,776	10	130,504	3	14.34	10	130,504	3	8.84
30	476,110	526,634	10	476,110	3	116.31	10	476,110	6	57.20
40	1,229,567	1,365,383	10	1,229,567	4	343.49	10	1,229,567	7	251.37

* These are the problems referred to in Table 3.

\+ This is the objective value of an arbitrary starting solution.

a - # of cuts generated; b - best recorded value; c - cut number at which best recorded solution was detected; d - cpu seconds of execution time on a CDC Cyber 74 computer

** Suboptimal solution value

References

1. Balas, E., "Intersection Cuts - A New Type of Cutting Planes for Integer Programming," Operations Research, Volume 19, pp. 19-39, 1971.

2. Balas, E., "Intersection Cuts from Disjunctive Constraints," Management Sciences Research Report, #330, Carnegie-Mellon Univ., February, 1974.

3. Balas, E., "Disjunctive Programming: Facets of the Convex Hull of Feasible Points," Management Sciences Research Report, #348, Carnegie-Mellon Univ., July, 1974.

4. Balas, E., and C. A. Burdet, "Maximizing a Convex Quadratic Function Subject to Linear Constraints," Management Sciences Research Report, #294, Carnegie-Mellon University, July, 1973.

5. Balinski, M. L. and A. Russakoff, "On the Assignment Polytope," SIAM Review, Volume 16, #4, October, 1974.

6. Burkard, R. E., and K. H. Stratmann, "Numerical Investigation on the Quadratic Assignment Problem," Naval Research Logistics Quarterly, Volume 25, #1, pp. 129-248, 1978.

7. Cabot, V. A. and R. L. Francis, "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," Operations Research, Volume 18, pp. 82-86, 1970.

8. Elshafei A. N., "Hospital Layout as a Quadratic Assignment Problem," Operational Research Quarterly, Volume 28, pp. 167-179, 1977.

9. Glover, F., "Convexity Cuts and Cut Search," Operations Research, Volume 21, pp. 123-134, 1974.

10. Glover, F., "Polyhedral Annexation in Mixed Integer and Combinatorial Programming," Mathematical Programming, Volume 8, pp. 161-188, 1975.

11. Graves, G. W., and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Management Science, Volume 17, pp. 453-471, 1970.

12. Jeroslow, R. G., "Cutting Plane Theory: Disjunctive Methods," Annals of Discrete Mathematics, Volume 1, pp. 293-330, 1977.

13. Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica, Volume 25, pp. 53-76, 1957.

14. Krarup, J., "Quadratic Assignment," Data, Volume 3, pp. 12-15, 1972.

15. Lawler, E. L., "The Quadratic Assignment Problem," Management Science, Volume 19, pp. 586-590, 1963.

16. Nugent, C. E., T. E. Vollmann and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," Operations Research, Volume 16, pp. 150-173, 1968.

17. Obata, T., and P. Mirchandani, "Algorithms for a Class of Quadratic Assignment Problems," presented at the Joint National TIMS/ORSA Meeting, New Orleans, May, 1979.

18. Pierce, J. F. and W. B. Crowston, "Tree Search Algorithms for Quadratic Assignment Problems," Naval Research Logistics Quarterly, Volume 18, pp. 1-36, 1971.

19. Roucairol, C., "A Reduction Method for Quadratic Assignment Problems," III Symposium Über Operations Research, Universität Manheim, September, 1978.

20. Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," SIAM Review, Volume 3, pp. 37-50, 1967.

21. Tui, H., "Concave Programming Under Linear Constraints," Soviet Mathematics, Volume 5, (English Translation), pp. 1437-1440, 1965.

ON THE USE OF EXACT AND HEURISTIC CUTTING PLANE METHODS FOR THE QUADRATIC ASSIGNMENT PROBLEM †

Mokhtar S. Bazaraa* and Hanif D. Sherali**

Abstract

This paper uses the formulation of the quadratic assignment problem as that of minimizing a concave quadratic function over the assignment polytope. Cutting plane procedures are investigated for solving this problem. A lower bound derived on the number of cuts needed for termination indicates that cutting plane procedures would require a huge computational effort for the exact solution of quadratic assignment problems. However, several heuristics which are derived from the cutting planes, produce optimal or good quality solutions early on in the search process.

---

† This research was supported under NSF grants #ENG77-07468 and #ENG79-08375

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

**Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

1. INTRODUCTION

This study addresses the use of cutting plane methods for solving quadratic assignment problems, which involve the assignment of m indivisible interacting facilities to m mutually exclusive locations. The problem was first formulated by Koopmans and Beckmann [14] and subsequently generalized and extended by Graves and Whinston [11], Lawler [16] and Pierce and Crowston [18]. Mathematically, defining x_{ij} to be one or zero according to whether or not facility i is placed at location j, for i, j = 1,...,m, one may write this problem as follows, where the superscript t denotes the transpose operation:

QAP 1: minimize $\left\{x^t Sx: x \in X_A = X \cap \{x: x \text{ is binary}\}\right\}$

where

$$X = \{x=(x_{11},\ldots,x_{mm}): \sum_{i=1}^{m} x_{ij}=1,\ j=1,\ldots,m,\ \sum_{j=1}^{m} x_{ij}=1,\ i=1,\ldots,m,\ x\geq 0\} \quad (1)$$

Here, S has components $s_{ijk\ell}$ which represent the interactive cost of simultaneously locating facility i at site j and facility k at site ℓ. We will assume that these components are all integral valued and that S is symmetric.

The formulation we will be concerned with is the one which may be obtained by transforming QAP 1 into a concave quadratic program. Observe that if we replace the objective function of QAP 1 with the function $x^t Dx$, where $D = S - IM$ (I being an identity matrix of size m^2 and M a scalar), the problem is essentially unchanged since $x^t IMx = mM$, a constant. Moreover, if one selects M to be larger than the greatest row-sum of S, then one can show that D is negative definite and hence that $x^t Dx$ is strictly concave, Finally, since the extreme points of X are in a one-to-one correspondence with points in X_A, and since the minimum of a strictly concave function over a bounded polyhedral set occurs at an extreme point, it follows that QAP 1 may be equivalently solved as:

QAP 2: minimize $\{x^t Dx: x \in X\}$

Note that local optimal solution to problem QAP 2 need not be globally optimal. Also, one may demonstrate an interesting fact about Problem QAP 2, namely that every extreme point is a Kuhn-Tucker solution. The proof of this is relatively straightforward and will be omitted.

The literature on quadratic assignment problems has thus far been devoted to basically two types of exact solution procedures. One technique is that of implicit enumeration [6, 11, 18] and the other is the use of some linearization scheme followed by the solution of a mixed integer program typically through Benders' decompostion [5, 13]. This motivated us to attempt using the rich literature available on cutting plane procedures in order to solve Problem QAP 2. We discovered that although the problem structure permitted the derivation of cuts stronger than normally available, the theoretical lower bound on the number of cuts required for termination was prohibitively large. Even though cutting planes seem to be ineffective for solving quadratic assignment problems exactly, as demonstrated in this paper, they can be used to produce efficient heuristics. This point is illustrated in the present paper.

2. DISCUSSION ON THE GENERATION AND CHARACTERISTICS OF SOME CUTTING PLANES

As an expedient in the discussion to follow, we will denote the vector x as

$$x = (x_{11},\ldots,x_{1m},x_{21},\ldots,x_{m1},\ldots,x_{mm}) \equiv (x_1,\ldots,x_m,x_{m+1},\ldots,x_{(m-1)m+1},\ldots,x_{m^2})$$

A typical cutting plane procedure would commence each iteration with a feasible point of X_A and generate a cut, or a valid inequality, which deletes this point but no other point of X_A with a quadratic objective value better than the current best. The procedure would then search for another point of X_A which is feasible to the cuts generated thus far. If none exists, then the current best solution would be declared optimal. Otherwise, a suitable improvement routine may be applied to the new feasible point, the current best solution updated if necessary, and the above procedure would then be repeated. This latter step of detecting a feasible solution to a system of cuts may be executed through an implicit enumeration scheme which is initialized just once and subsequently updated at each iteration. The feasible solution thus obtained may be improved, if possible, through pairwise interchanges which preserve feasibility. We will now devote our attention to the generation of valid inequalities based on an available feasible point of X_A.

Intersection Cuts

Let us begin by briefly discussing Tui's [21] cutting plane method with some minor modifications. Suppose we have a point $\bar{x}$ of X_A feasible to the cuts generated thus far. Consider an extended simplex tableau yielding a basic representation of $\bar{x}$ in terms of only the constraints in X and not including any cutting planes. Correspondingly, let J denote the set of nonbasic variables and note that $|J| = (m-1)^2$. Identify the $(m-1)^2$ edges incident at $\bar{x}$, each edge associated with a single nonbasic variable, and let $\bar{a}^j$ be the extended column of the nonbasic variable x_j, $j\varepsilon J$, in the tableau under consideration. Hence, we may write any $x\varepsilon X$ as

$$x = \bar{x} - \sum_{j\varepsilon J} \bar{a}^j \lambda_j, \ \lambda_j \geq 0 \text{ for each } j\varepsilon J \tag{2}$$

Further, define a halfline corresponding to each edge incident at $\bar{x}$ according to

$$\xi^j = \{x: x = \bar{x} - \bar{a}^j \lambda_j, \ \lambda_j \geq 0\} \text{ for each } j\varepsilon J \tag{3}$$

Now, let ν be the current best objective function value for Problem QAP 2 and let $\bar{\nu} = \nu - 1$. Consider the level set

$$L(\bar{\nu}) = \{x: \ x^t Dx \geq \bar{\nu}\} \tag{4}$$

Note that $L(\bar{\nu})$ is a convex set which contains $\bar{x}$ in its interior and which does not contain in its interior any point of X_A which has an objective function value less than or equal to $\bar{\nu}$. Hence a valid intersection cut may be derived from $L(\bar{\nu})$ as in [1] or [9]. This cut is defined by a hyperplane passing through the $(m-1)^2$ distinct points of intersection of the halflines ξ^j, $j\varepsilon J$, with $L(\bar{\nu})$ and is given by

$$\sum_{j\varepsilon J} x_j/\bar{\lambda}_j \geq 1 \tag{5}$$

where

$$\bar{\lambda}_j = \sup \{\lambda_j: \ \bar{x} - \bar{a}^j \lambda_j \subset L(\bar{\nu})\}, \text{ for each } j\varepsilon J \tag{6}$$

We next demonstrate that $\bar{\lambda}_j$ is a finite positive scalar for each $j\varepsilon J$. Note that, for each $j\varepsilon J$, $\bar{\lambda}_j$ is obtained as a solution λ_j to the equation

$$(\bar{x}-\lambda_j \bar{a}^j)^t D (\bar{x}-\lambda_j \bar{a}^j) = \bar{\nu}$$

This yields:

$$d_{1j}\lambda_j^2 + d_{2j}\lambda_j - d_3 = 0 \tag{7}$$

where,

$$d_{1j} = -(\bar{a}^j)^t D \bar{a}^j > 0,\ d_{2j} = 2(\bar{a}^j)^t D \bar{x}, \text{ and}$$

$$d_3 = \bar{x}^t D \bar{x} - \bar{\nu} > 0 \text{ for each } j\varepsilon J \tag{8}$$

Thus, $\bar{\lambda}_j$ is the positive root of (7), or

$$0 < \lambda_j = (-d_{2j} + \sqrt{d_{2j}^2 + 4d_{1j}d_3})/2d_{1j} < \infty, \text{ for each } j\varepsilon J \tag{9}$$

The cut given by (5) can be modified as follows. Let us partition the nonbasic variable set J into two disjoint sets J_d and J_{nd}, which respectively represent the index sets of nonbasic variables that lead to degenerate and nondegenerate pivots on the basis under consideration. Now, note that for $j\varepsilon J_{nd}$, one may assume that $(1/\bar{\lambda}_j) < 1$ for if $\bar{\lambda}_j \leq 1$, then from (3), the extreme point of X adjacent to $\bar{x}$ given by $\bar{x}-\bar{a}^j$ has value $\leq \bar{\nu}$. Hence one may abort the current cut and generate a new cut from this improved solution, which must necessarily be feasible. Hence, when a cut is finally generated, we must have:

$$1/\bar{\lambda}_j < 1 \quad \text{for each } j \ \varepsilon \ J_{nd} \tag{10}$$

Now, consider zero-one solutions in relation to the cuts

$$\sum_{j\varepsilon J_d} x_j/\bar{\lambda}_j + \sum_{j\varepsilon J_{nd}} x_j/\bar{\lambda}_j \geq 1 \tag{11}$$

and

$$\sum_{j\varepsilon J_d} 2x_j + \sum_{j\varepsilon J_{nd}} x_j \geq 2 \tag{12}$$

We will now validate the cut (12). In order to do this, it is sufficient to show that any zero-one solution deleted by (12) is also infeasible to (11). Particularly, if (12) is violated, then $x_j = 0$ for each $j\varepsilon J_d$ and $\sum_{j\varepsilon J_{nd}} x_j \leq 1$. This further implies that either $\sum_{j\varepsilon J_{nd}} x_j = 0$ in which case (11) is violated or else $\sum_{j\varepsilon J_{nd}} x_j = 1$. In this latter case, and noting from (10) that $1/\bar{\lambda}_j < 1$ for each $j\varepsilon J_{nd}$, it follows that $\sum_{j\varepsilon J_{nd}} x_j/\bar{\lambda}_j < 1$, again violating (11). This shows that the cut given by (12) is indeed valid.

It turns out that the cut (12) itself may be further strengthened. Observe that zero-one points feasible to (12) which have exactly one variable equal to unity do not correspond to extreme points of X since they correspond to degenerate pivots. Hence, extreme points of X feasible to (12) must have at least two nonbasic variables equal to unity. In other words, the cut (12) can be strengthened to:

$$\sum_{j\varepsilon J} x_j \geq 2 \tag{13}$$

Note that the cut (13) is numerically well conditioned. More importantly, it avoids completely the computation of $\bar{\lambda}_j$ for $j\varepsilon J_d$. The computation of $\bar{\lambda}_j$ for $j\varepsilon J_{nd}$ is, however, required in order to verify (10). Moreover, for $j\varepsilon J_{nd}$, the following relationship can be derived from (9):

$$1/\bar{\lambda}_j < 1 \iff d_{1j} + d_{2j} < d_3 \tag{14}$$

Thus if $d_{1j} + d_{2j} \geq d_3$ for some $j \in J_{nd}$, an improved extreme point is obtained by entering x_j in the basis. If (14) holds for each $j \in J_{nd}$, then the cut (13) is valid.

Disjunctive Cuts

A linear disjunction is a logical statement which asserts that at least p of q linear inequality systems must be satisfied. We state two such disjunctions and derive appropriate cuts therefrom. The material which follows is self-contained, and a reader interested in disjunctive programming principles is referred to [2, 3, 10, 12].

To begin with, consider an extreme point $\bar{x}$ of X feasible to the cuts generated thus far and suppose that $\bar{x}_{i,a(i)} = 1$ for $i = 1,\ldots m$. Note that any other extreme point of X has at least two of the variables $x_{i,a(i)}$, for $i = 1,\ldots,m$, equal to zero. Further, suppose that $\bar{x}$ is such that any pairwise interchange on it results in either a non-improving solution or a solution infeasible to the cuts generated thus far. We may hence assert the following disjunction:

DC 1 At least three of the variables $x_{i,a(i)}$ must be equal to zero. That is, at least 3 of the following m inequality systems must be satisfied

$$\{x_{i,a(i)} \leq 0,\ x \geq 0\},\ i = 1,\ldots m$$

A strongest, valid disjunctive cut based on DC 1, that is, one which deletes all points not satisfying DC 1, but none satisfying this statement, is easily obtained as

$$\sum_{i=1}^{m} x_{i,a(i)} \leq m - 3 \tag{15}$$

We will now state a second disjunction, DC 2, based on the objective function of Problem QAP 2 as opposed to DC 1 which is based on the constraints of this problem. Toward this end, consider the reverse polar set

$$X^0(\bar{\nu}) = \{x: \; x^t D x^r \geq \bar{\nu} \text{ for each } x^r \varepsilon X_A\} \tag{16}$$

where $\bar{\nu} = \nu - 1$, and ν is known current best objective function value of Problem QAP 2. Clearly, the interior of $X^0(\bar{\nu})$ contains no extreme point of X with an objective function value less than $\bar{\nu}$. We may hence assert the following disjunction:

DC 2 At least one of the following m! linear inequality systems must be satisfied

$$\{\sum_{i=1}^{m} \sum_{j=1}^{m} (\sum_{k=1}^{m} \sum_{\ell=1}^{m} d_{ijk\ell} x^r_{k\ell}) \; x_{ij} \leq \bar{\nu}, \; x \geq 0\}, \; r = 1,\ldots,m!, \; x^r \varepsilon X_A$$

Hence, for DC 2 to be satisfied, since $x \geq 0$, we must have,

$$\hat{u}^t x \equiv \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\underset{x^r \varepsilon X_A}{\text{minimum}} \; \sum_{k=1}^{m} \sum_{\ell=1}^{m} d_{ijk\ell} x^r_{k\ell} \right] x_{ij} \leq \bar{\nu} \tag{17}$$

It is easy to see from the definition of D and M that for points in X_A, the cut (17) is identical to the cut

$$u^t x \equiv \sum_{i=1}^{m} \sum_{j=1}^{m} \left[\underset{\substack{x^r \varepsilon X_A \\ x^r_{ij}=1}}{\text{minimum}} \; \sum_{k=1}^{m} \sum_{\ell=1}^{m} s_{ijk\ell} x^r_{k\ell} \right] x_{ij} \leq (\bar{\nu} + mM) = \bar{\bar{\nu}}, \text{ say} \tag{18}$$

In other words, $u^t x \leq \bar{\bar{\nu}}$ is a valid inequality for DC 2. It is interesting to note here that $u^t x$ is precisely the linear bounding expression derived by Cabot and Francis [7] for the quadratic assignment problem, using a different relaxation argument. Observe that the cut (17) depends only on the current best objective function value and is not derived from any particular point in X_A.

To summarize, the cut (15) may be generated each time a feasible point $\bar{x}\varepsilon X_A$ is located such that pairwise interchanges on $\bar{x}$ lead to non-improving or infeasible points. Further, the cut (18) has its right-hand-side updated each time an improved solution is detected.

Number of Cuts Needed

Now observe that the cut (15) deletes precisely $1 + \frac{m(m-1)}{2}$ points of X_A. Namely, it deletes the current point $\bar{x}$, as well as those points in X_A which are generated through pairwise exchanges on $\bar{x}$. Any other point in X_A must necessary have strictly less than (m-2) of the variables $x_{i,a(i)}$ equal to one. Further, recall that (13) deletes $\bar{x}$ and only those points in X_A which are obtained via a single nondegenerate pivot on $\bar{x}$, for any other point in X_A must have at least two of the variables x_j, $j\varepsilon J$ equal to one. Now, it can be shown that the maximum number of nondegenerate pivots from any basis representing $\bar{x}$ is $m(m-1)/2$. Hence, it follows that $\bar{x}$ also deletes at most $1 + m(m-1)/2$ points in X_A. Consequently, a lower bound on the number of cuts of the type (13) or (15) needed for termination is:

$$\#\text{ of cuts} \geq \frac{m!}{1 + \frac{m(m-1)}{2}} \approx 2(m-2)! \qquad (19)$$

Now, consider the cut (18). Computationally, we found that for most problems with $m \geq 7$, $\underset{x\varepsilon X_A}{\text{maximum}}\ u^t x < \hat{\nu}$, where $\hat{\nu}$ is the best known objective value of QAP 2. This essentially states that the cut (18) does not delete any point of X_A for even the smallest known valid value of $\bar{\bar{\nu}}$.

In view of the above considerations, we found it futile to use cutting plane procedures to solve quadratic assignment problems exactly. We merely remark here that we attempted other cutting planes like that derived from reverse outer polar sets [4], another derived through a linear bounding scheme available from Roucairol's reduction method [19], and we also attempted a direct application of Tui's [21] procedure, all in vain. Assuming that the magnitude of M in the objective of QAP 2 was having an adverse effect on the depth of the cuts, we even devised a technique to find out the largest integer q for which the cut $\sum_{j \varepsilon J} x_j \geq q$ is valid for Problem QAP 1. Empirically, we were never able to improve on the cut (13).

3. HEURISTICS DERIVED FROM THE CUTTING PLANE PROCEDURES

Our study of cutting planes for the quadratic assignment problem led us to observe two facts which seem to lend themselves to the development of good heuristic schemes. Firstly, consider the cut (18) and define

$$\nu_{min} = \{\text{minimum } u^t x: x \varepsilon X_A\}, \ \nu_{max} = \text{maximum } \{u^t x: \ x \varepsilon X_A\} \qquad (20)$$

Using several test problems available in the literature, we found that $\bar{\nu} = u^t x^*$ occurs in the earlier protion of the range $[\nu_{min}, \nu_{max}]$, where x^* is either the known optimal or the best reported solution. Table 1 below presents some characteristics of the this cut and indicates that $\bar{\nu}$ is typically contained in the top 15% of the range $[\nu_{min}, \nu_{max}]$.

Table 1. Some Characteristics of the Disjunctive Cost Cut (18)

Problem	m	ν_{min}	ν_{max}	$x^{*t}Sx^{*}$	$\bar{\nu} = u^t x^*$	$\frac{\bar{\nu} - \nu_{min}}{\nu_{max} - \nu_{min}}$
Nugent, Vollman and	5	50	55	50	50	0.00
Ruml's Problems [17]	6	82	84	86	82	0.00
	7	137	144	148	140	0.43
	8	186	199	214	188	0.15
	12	493	517	578	496	0.13
	15	963	1,034	1,150	970	0.09
	20	2,057	2,243	2,570	2,070	0.07
	30	4,539	4,948	6,154	4,587	0.12
Steinberg's Problems [20]:						
Squared Euclidean Distance	36	8,653	14,305	15,852	8,899	0.04
Rectilinear Distance	36	7,124	8,590	9,604	7,198	0.05
Elshafei's Problem [9]	19	11,971,949	25,541,722	17,212,548	12,568,503	0.04
Krarup's Problem [15]	32	67,390	77,680	90,220	69,140	0.17

The above observation is incorporated into a heuristic procedure discussed below for the quadratic assignment problem. Another useful device which we employed is the use of a fictitious linear assignment problem for generating a new assignment solution. Particularly, suppose that c cuts of the form $\alpha_i^t x \geq \theta_i$ for $i=1,\ldots,c$ have already been generated. A point $x \epsilon X_A$ that satisfies these cuts is sought. If suitable weights $w_1,\ldots,w_c$ are chosen, the fictitious linear assignment problem to maximize $\sum_{i=1}^{m} w_i\alpha_i^t x$ subject to $x\varepsilon X_A$ often produced a solution which is feasible to the cuts. The weight w_i used is given by $1/|\text{maximum } \{\alpha_i^t x : x\varepsilon X_A\}|$ so that each cut is normalized by dividing it with its maximum absolute value. The fictitious objective function thus seeks a point in X_A which is roughly equidistant from each cut. We also found it quite helpful to add the disjunctive cost cut into the objective function via a suitable weight w leading to the following problem:

P: maximize $\beta^t x$

subject to $x\varepsilon X_A$

$$\text{where } \beta = \sum_{i=1}^{c} w_i\alpha_i - \left(\frac{w}{\nu_{min}}\right) u \qquad (21)$$

Figure 1 gives a flow chart for the proposed heuristic schemes using either cuts (13) or (15). Both these procedures incorporate (21). Further, the cuts are used only to update β and need not be stored. Also an improvement routine using pairwise exchanges on the solution of Problem P is incorporated. If the pairwise exchange scheme leads to an improved solution, the latter is used to generate the next cut. Otherwise, the next cut is generated from the solution to P. When the disjunctive cut (15) is used, one needs to verify that no single pairwise exchange improves the solution over the current best value at a point from which the cut is being generated. In case of the intersection cut (13), if $1/\bar{\lambda}_j \geq 1$ for $j\varepsilon J_{nd}$, then the corresponding improved solution found is adopted as discussed in Section 2.

Finally, note that the expression $u^t x$ is used in the fictitious problem P via a weighting factor w/ν_{min}. By suitably adjusting w, this problem P may be used to generate solutions x in X_A for which $u^t x$ lies in the initial portion of the range $[\nu_{min}, \nu_{max}]$. As discussed earlier, such solutions x are likely to have a good (quadratic) objective function value. The strategy we adopted was to initialize w=1 and then increment it by one every five cuts. Further, to obtain a starting solution, we arranged the facilities in nonincreasing order of their flow sums and the locations in nondecreasing order of their distance sums, and then matched the two ordered sets pointwise. Thereafter, we performed pairwise exchanges until no further improvement was obtained, and used the resulting solution to initialize our heuristic.

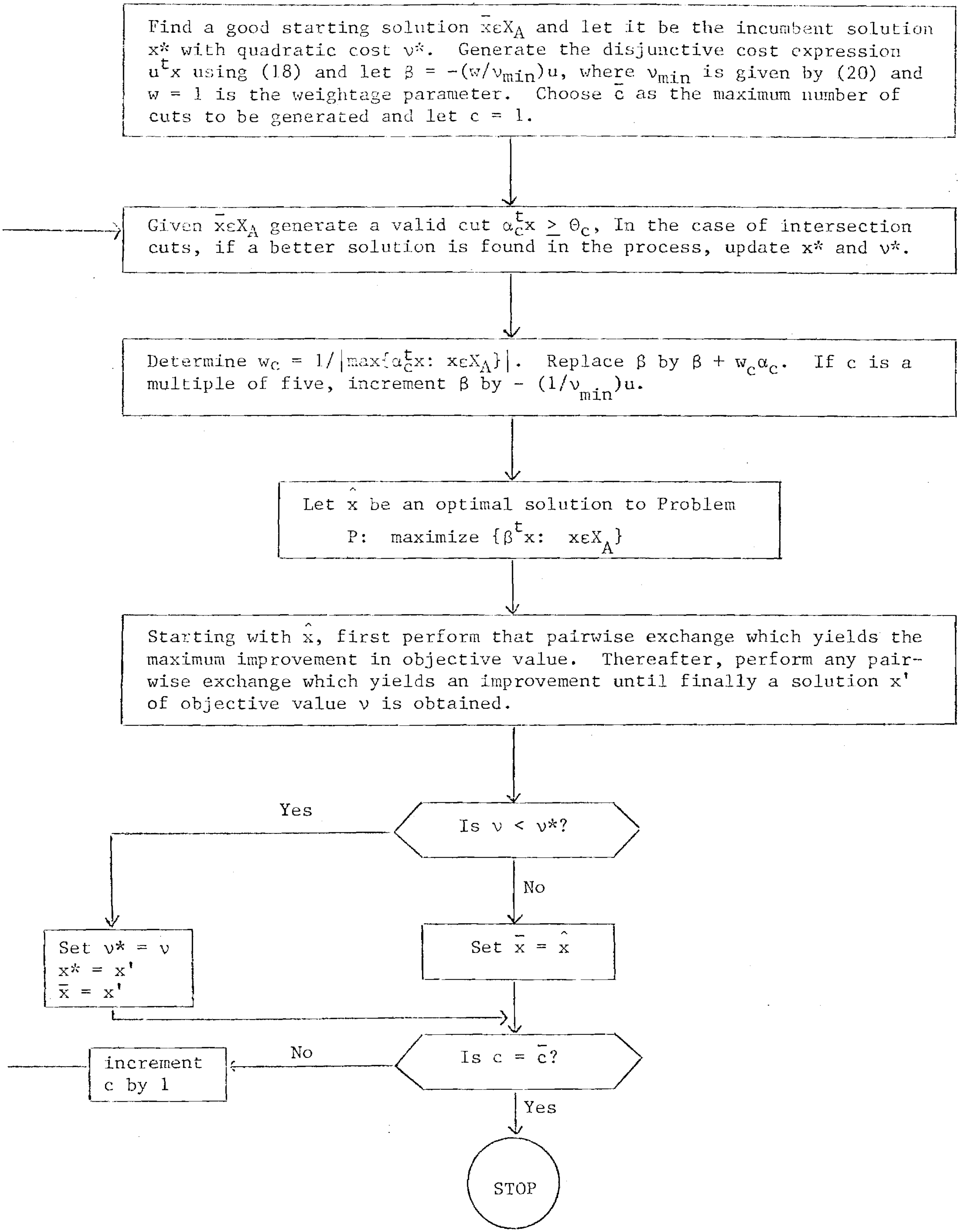

FIGURE 1. Flow Chart for Heuristic Procedures Employing Intersection Cuts or Disjunctive Cuts

4. COMPUTATIONAL TESTING

In this section, we report our computational experience using test problems available in the literature.

Table 2 presents our results with test problems taken from references [8, 15, 19,20]. For both the cutting plane heuristics, we employed the starting solution obtained by matching ordered flow and distance vectors and then performing pairwise interchanges on the resulting solution. Column 'a' gives the starting solution values, column 'b' gives the best values obtained on using this initial solution and terminating the procedure after 50 cuts. Column 'c' gives the cut indices at which these best solutions were found. Column 'd' gives the execution time in cpu seconds for a run which generates 50 cuts on a CDC Cyber 70 Model 74-28/CDC 6400 computer, with coding in FORTRAN IV. These times do not include the effort for generating either the starting solution or the expression $u^t x$ of Equation (18).

Through a few subsequent runs employing better quality starting solutions as obtained over previous runs of either procedures, we were able to use these heuristics to further improve upon the initial run solutions. Column 'e' gives the best objective values we were able to obtain in this manner. For each problem the heuristic produced the best known solution available in the literature. Finally, column 'f' gives the locations of facilities 1,...,m respectively, corresponding to the solutions with objective values given in column 'e'.

Table 2. Computational Results for the Cutting Plane Heuristics

Problem	m	a	Intersection Cuts (13)			Disjunctive Cuts (15)			e	f
			b	c	d	b	c	d		
Nugent, Vollmann and Ruml's Problems [17]	5	26	25	2	1.33	25	4	0.60	25	3,4,5,1,2
	6	43	43	1	2.61	43	1	1.00	43	3,2,1,6,5,4
	7	78	74	3	4.49	74	2	1.03	74	5,6,7,1,2,3,4
	8	118	107	4	7.13	107	3	4.16	107	3,4,8,2,1,5,6,7
	12	309	289	3	12.9	289	7	10.92	289	5,1,9,8,4,3,11,7,10,2,6,12
	15	610	575	6	25.65	575	6	15.30	575	15,14,9,10,2,3,7,12,11,5,6,1,13,8,4,
	20	1,334	1,290	8	70.88	1,285	29	34.91	1,285	17,9,2,10,19,16,18,12,1,3,7,8,11,4, 14,6,20,5,15,13
	30	3,185	3,089	16	261.67	3,078	10	198.39	3,077	11,2,20,26,1,3,16,22,8,10,29,4,9,25, 30,21,12,23,15,19,7,17,24,5,18,6,28, 13,14,27
Steinberg's Problems [20] Squared Euclidean Distances	36	8,467	7,946	15	542.83	8,291	8	580.01	7,926	4,19,29,21,30,31,13,20,2,12,32,23,22 24,3,1,10,11,15,14,34,35,25,36,26,27 33,5,6,7,8,16,18,17,9,28
Steinberg's Problems [20] Rectilinear Distances	36	5,213	5,019	24	422.00	4,914	44	514.74	4,802	23,18,8,16,7,6,24,17,34,25,5,14,15,4 33,35,27,26,21,13,11,2,12,1,10,19,3, 22,32,31,30,20,28,29,36,9
Elshafei's Problem [8]	19	c_1	c_2	5	73.36	c_2	2	37.90	c_2	17,18,19,11,12,9,3,14,1,2,10,13,7,5 15,16,8,4,6
Krarup's Problem [15]	32	47,360	45,485	41	354.11	45,490	16	258.46	45,110	31,26,25,22,23,21,7,11,12,3,24,20,8, 16,18,27,28,32,1,10,6,19,2,9,14,30, 15,4,5,29,13,17

a, b, c, d, e, f: See text of Section 4 for connotation.

$c_1 = 11299400$

$c_2 = 8606274$

REFERENCES

1. Balas, E., "Intersection Cuts - A New Type of Cutting Planes for Integer Programming," Operations Research, Volume 19, pp. 19-39, 1971.

2. Balas, E., "Intersection Cuts from Disjunctive Constraints," Management Sciences Research Report, #330, Carnegie-Mellon University, Feb. 1974.

3. Balas, E., "Disjunctive Programming: Facets of the Convex Hull of Feasible Points," Management Sciences Research Report, #348, Carnegie-Mellon University, July 1974.

4. Balas, E. and C. A. Burdet, "Maximizing a Convex Quadratic Function Subject to Linear Constraints, "Management Sciences Research Report, #294, Carnegie-Mellon University, July 1973.

5. Bazaraa, M. S. and H. D. Sherali, "Benders' Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem," Naval Research Logistics Quarterly, Volume 27, No. 1, pp. 29-42, 1980.

6. Burkard, R. E. and K. H. Stratmann, "Numerical Investigation on the Quadratic Assignment Problem," Naval Research Logistics Quarterly, Volume 25, No. 1, pp. 229-248, 1978.

7. Cabot, V. A. and R. L. Francis, "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," Operations Research, Volume 18, pp. 82-86, 1970.

8. Elshafei, A. N., "Hospital Layout as a Quadratic Assignment Problem," Operational Research Quarterly, Volume 28, pp. 167-179, 1977.

9. Glover, F., "Convexity Cuts and Cut Search," Operations Research, Volume 21, pp. 123-134, 1974.

10. Glover, F., "Polyhedral Annexation in Mixed Integer and Combinatorial Programming," Mathematical Programming,Volume 8, pp. 161-188, 1975.

11. Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Management Science, Volume 17, pp. 453-471, 1970.

12. Jeroslow, R. G., "Cutting Plane Theory: Disjunctive Methods, "Annals of Discrete Mathematics, Volume 1, pp. 293-330, 1977.

13. Kaufman, L. and F. Broeckx, "An Algorithm for the Quadratic Assignment Problem Using Benders' Decomposition," European Journal of Operational Research, Volume 2, pp. 204-211, 1978.

14. Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica, Volume 25, pp. 53-76, 1957.

15. Krarup, J., "Quadratic Assignment," Data, Volume 3, pp. 12-15, 1972.

16. Lawler, E. L., "The Quadratic Assignment Problem," Management Science, Volume 19, pp. 586-590, 1963.

17. Nugent, C. E., T. E. Vollman and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," Operations Research, Volume 16, pp. 150-173, 1968.

18. Pierce, J. F. and W. B. Crowston, "Tree Search Algorithms for Quadratic Assignment Problems," Naval Research Logistics Quarterly, Volume 18, pp. 1-36, 1971.

19. Roucairol, C., "A Reduction Method for Quadratic Assignment Problems," III Symposium Über Operations Research, Universität Manheim, September 1978.

20. Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," SIAM Review, Volume 3, pp. 37-50, 1967.

21. Tui, H., "Concave Programming Under Linear Constraints, " Soviet Mathematics Volume 5, (English Translation), pp. 1437-1440, 1965,

A Branch and Bound Based Heuristic for Solving the Quadratic Assignment Problem

M. S. Bazaraa and O. Kirca

Abstract

In this paper a branch and bound algorithm is proposed for solving the quadratic assignment problem. Using symmetric properties of the problem, the algorithm eliminates "mirror image" branches, thus reducing the search effort. Several routines that transform the procedure into an efficient heuristic are also implemented. These include certain 2-way and 4-way exchanges, selective branching rules, and the use of variable upper bounding techniques for enhancing the speed of fathoming.

The computational results are quite encouraging. As an exact scheme, the algorithm solved the 12 facility problem of Nugent et al and the 19 facility problem of Elshafei. More importantly, as a heuristic, the procedure produced the best known solutions for all well-known problems in the literature, and produced improved solutions in several cases.

This research is supported under NSF grant ENG-79-08375 and ONR grant N00014-80-k-0709.

1. INTRODUCTION

The quadratic assignment problem, as given by Koopmans and Beckmann (1957), can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} f_{ik} d_{j\ell} x_{ij} x_{k\ell} \tag{1}$$

$$\text{Subject to:} \quad \sum_{i=1}^{m} x_{ij} = 1 \qquad j=1,\ldots,m \tag{2}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i=1,\ldots,m \tag{3}$$

$$x_{ij} = 0 \text{ or } 1 \qquad i,j=1,\ldots,m \tag{4}$$

The above problem can be interpreted as follows. There are m indivisible objects to be assigned to m indivisible locations, where f_{ik} is the flow or interaction between objects i and k and $d_{j\ell}$ is the distance between locations j and ℓ. The objective is to assign the objects to the locations such that the sum of pairwise interactions among objects weighed by the distance between their respective locations is minimized. Without loss of generality it is assumed that the interaction and distance matrices are symmetric.

There exists two approaches for solving the quadratic assignment problem exactly. The first approach utilizes the concept of branch and bound or implicit enumeration, as in the works of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Bazaraa and Elshafei (1979), Burkard and Stratmann (1978), Roucairol (1978), Pierce and Crowston (1971), Land (1963), and Gavett and

Plyter (1966). Secondly, through an appropriate transformation, the problem can be reformulated as a linear mixed-integer program which is solved by cutting planes or by a suitable mixed-integer programming package. The algorithms of Bazaraa and Sherali (1980), Kaufman and Broeckx (1978), and Love and Wong (1976) fall into this class.

Due to the complexity of the quadratic assignment problem, in general, none of the above methods can solve problems with dimension $m > 15$ effectively. Thus for larger problems, a considerable amount of effort has been given to the development of inexact methods that obtain good quality solutions with a reasonable computational effort. A comprehensive survey of inexact methods can be found in the works of Sherali (1979) and Burkard and Stratmann (1978). A summary of inexact methods for solving the quadratic assignment problem is given below.

a) Construction Methods

Starting with a partial solution or the null assignment, a complete assignment is reached iteratively by locating one or more objects at each iteration.

b) Improvement Methods

Starting with a complete assignment of objects, an improvement over the incumbent objective function value is sought by interchanging the locations of several objects. The procedure is terminated when no further improvements are possible.

c) Hybrid Methods

Methods in this class combine several features of exact and inexact procedures.

According to the computational experience reported in the literature, it seems that hybrid methods are emerging as the most successful approach for solving large quadratic assignment problems. Examples of such procedures are the methods of Bazaraa and Sherali (1980) and Burkard and Stratmann (1978). Both methods use an exact solution scheme in conjunction with some improvement procedures. Bazaraa and Sherali implement Benders' partitioning method to a mixed-integer formulation of the problem and apply several improvement procedures to the solutions found throughout the course of partitioning. The method of Burkard and Stratmann alternates between a branch and bound (Perturbation) routine and an exchange routine (Verbes) until no better solutions can be obtained.

In this paper, a branch and bound algorithm for solving the quadratic assignment problem is proposed. The main feature of the procedure is the elimination of "mirror image" branches in the search tree. The branch and bound procedure is modified in order to accelerate the computations resulting in an efficient heuristic procedure with the following characteristics:

1. Several improvement routines are used in conjunction with the branch and bound scheme. The extent of using these improvement routines is a function of the branch and bound tree level.

2. Several heuristics are utilized to eliminate the search effort at branches which are likely not to lead to objective value improvements. Furthermore, variable upper bounds are used to reduce the number of solutions examined.

The computational results are quite encouraging. As an exact procedure, the algorithm solved the 12 facility problem of Nugent et al (1968) and the

19 facility problem of Elshafei (1977). More importantly, as an inexact procedure, the modified branch and bound algorithm produced the best known or improved solutions for all well-known problems in the literature of the quadratic assignment problem.

2. AN EXACT BRANCH AND BOUND PROCEDURE

Branch and bound procedures for the quadratic assignment problem can be classified into single assignment algorithms, pair assignment algorithms, and pair exclusion algorithms. At each stage of a single assignment algorithm, one unassigned object is assigned to an unoccupied location. The procedures of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Burkard and Stratmann (1978), and Bazaraa and Elshafei (1979) are some examples of single assignment algorithms. The pair assignment algorithms proceed by simultaneously locating two unassigned objects to two vacant locations. The procedures proposed by Land (1963) and Gavett and Plyter (1966) are of this type. Pierce and Crowston (1971) proposed a pair exclusion procedure where the algorithm proceeds on the basis of a stage-by-stage exclusion of assignments from a solution to the problem.

The proposed procedure is of the single assignment type where the following general approach is pursued. Let:

$X = \{x\text{: } x \text{ satisfies (2), (3), (4)}\}$

I = set of assigned objects

$\bar{I}$ = complement of I, that is, set of unassigned objects

$\sigma(i)$ = location to which object $i \varepsilon I$ is assigned

$J = \{\sigma(i)\text{: } i \varepsilon I\}$

$\bar{J}$ = complement of J, that is, set of vacant locations

$P = (I,J) = \{(i,\sigma(i))\text{: } i \varepsilon I\}$

$X_P = \{x\text{: } x \varepsilon X \text{ and } x_{ij} = 1 \text{ for all } (i,j) \varepsilon P\}$

μ^* = upper bound on the value of the objective function

π^* = assignment vector of the objects corresponding to the upper bound μ^*, $\pi^* \varepsilon X$.

At each stage of the procedure, we select a partial assignment of objects I and locations J that form the partial assignment set P. The set X_P is then partitioned into $x_{P_1}, x_{P_2}, \cdots, x_{P_n}$ such that:

$$X_{P_k} \cap X_{P_\ell} = \emptyset \qquad \text{if } k \neq \ell; \quad k, \ell = 1, \ldots, n$$

$$\bigcup_{k=1}^{n} X_{P_k} = X_P$$

For a selected partial assignment P_k, a lower bound Z_{P_k} is computed. If $Z_{P_k} \geq \mu^*$ then P_k is fathomed, that is, discarded from further considerations. Otherwise, a complete assignment π_{P_k} is sought and its corresponding objective value μ_{P_k} is calculated. If $\mu_{P_k} < \mu^*$ then μ^* and π^* are updated to μ_{P_k} and π_{P_k}, respectively. The above procedure is repeated until no partial assignment P whose lower bound is less than μ^* can be found.

The process of partitioning the X_P into $X_{P_1}, X_{P_2}, \cdots, X_{P_n}$ is referred to as <u>branching</u> from the <u>node</u> representing the partial assignment P. The number of objects in the partial solution is called the <u>level</u> of the tree. The <u>active nodes</u> or <u>active branches</u> is the set of all partial solutions that have not been fathomed or selected for further branching. A branch and bound scheme for solving the quadratic assignment problem can be fully described by specifying rules for:

1) Computing a lower bound Z_P on the objective value of all completions of a partial solution P.

2) Choosing an active node (partial solution) for branching.

3) Branching from a selected partial assignment.

There exist several lower bounding procedures such as those of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Roucairol (1978), Edwards (1980), Christofides et al (1980) and Frieze and Yadegar (1981). Considering the strength of the bounds and the computational effort, the procedure of Gilmore-Lawler seems to be the most effective. This procedure is adopted here and is described briefly as follows.

Given a partial assignment $P = (I,J)$, the lower bound Z_p is obtained by solving the following linear assignment problem LAP:

$$Z_P = \underset{x \in X_P}{\text{Minimum}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} w_{ij} x_{ij} + \nu_P \qquad (6)$$

where:

$$w_{ij} = 2 \sum_{k \in I} f_{ik} d_{j\sigma(k)} + < \bar{f}(i), \bar{d}(j) >$$

$\bar{f}(i)$ = vector of interactions of object i with other unassigned objects in $\bar{I}$, where the elements of the vector are sorted in an ascending order.

$\bar{d}(j)$ = vector of distances from location j to other unoccupied locations in $\bar{J}$, where distances are sorted on a descending order.

$$\nu_P = \sum_{i \in I} \sum_{k \in I} f_{ik} d_{\sigma(i)\sigma(k)}$$

$<\cdot, \cdot>$: stands for the inner product of two vectors.

In the above linear assignment problem, w_{ij} is a lower bound on the assignment of object $i \in \bar{I}$ to location $j \in \bar{J}$. The fixed cost ν_p is the value accruing from the current assignment of objects in I to locations in J. Let the optimal assignment of objects in problem LAP be a(i) for $i\varepsilon\bar{I}$. Then:

1) Z_p is a lower bound on the objective value of all completions of the partial assignment P.

2) The quadratic cost μ_p of the solution π_p given below can be used to update μ^* provided that $\mu_p < \mu^*$.

$$\pi_p(i) = \begin{cases} \sigma(i) & \text{if } i\varepsilon I \\ a(i) & i\varepsilon\bar{I} \end{cases}$$

3. At optimality of problem LAP, a set of Lagrangian multipliers u_i for $i\varepsilon\bar{I}$ and v_j for $j\varepsilon\bar{J}$ with the following properties, is available:

$$w'_{ij} = w_{ij} - u_i - v_i \geq 0 \quad i\varepsilon\bar{I},\ j\varepsilon\bar{J}$$

$$w'_{ij} = 0 \qquad \text{if} \quad x_{ij} = 1$$

The reduced costs w'_{ij} for $i\varepsilon\bar{I}$, $j\varepsilon\bar{J}$ can be utilized to compute lower bounds for all branches emanating from the node associated with the partial solution P = (I,J) without the need for solving new linear assignment problems. This procedure is called the <u>alternative cost method</u> and has been applied by Little et al (1963) for the travelling salesman problem and later used by Pierce and Crowston (1971) for the quadratic assignment problem.

To demonstrate the use of the alternative cost principle, consider the partial assignment P = (I,J). Let b(j) be the object assigned to location $j\varepsilon\bar{J}$ in the solution to problem LAP. Now consider the branch with the partial assignment

$$P' = P \cup \{(r,s)\} \quad \text{for} \quad r \varepsilon \bar{I} \text{ and } s \varepsilon \bar{J}$$

A lower bound $\bar{Z}_{P'}$ on the objective values of all completions of P' is readily available as:

$$\bar{Z}_{P'} = Z_P + \gamma_{rs}$$

where:

$$\gamma_{rs} = \begin{cases} 0 \text{ if } \quad s = a(r) \\ \text{minimum } \{w'_{rs} + w'_{b(s)\ a(r)},\ w'_{rs} + \alpha_{b(s)} + \beta_{a(r)}\} \end{cases}$$

$$\alpha_i = \underset{\substack{\ell \varepsilon \bar{J} \\ \ell \neq a(i)}}{\text{minimum}}\ w'_{i\ell}$$

$$\beta_j = \underset{\substack{\ell \varepsilon \bar{I} \\ \ell \neq b(\bar{j})}}{\text{minimum}}\ w'_{\ell j}$$

2.1 Selection of the Branching Node

At each stage of the branch and bound procedure, a partial assignment has to be selected among all active branches. The following two strategies are typically used:

1) Depth First

Choose the active branch with the least lower bound among the most recently created active branches.

2) Breadth First

Choose the active branch with the least lower bound among all active branches in the current decision tree.

The attainment of good quality solutions early on is of great importance for the quadratic assignment problem, especially if the algorithm is eventually used as a heuristic. Implementing the depth or breadth strategies alone is not satisfactory. The correlation between lower bounds and quality of partial assignments at low levels of the branch and bound tree is not strong. Thus it is highly likely that a depth strategy may select poor quality branches to pursue initially so that good quality solutions are obtained only after a large number of nodes is evaluated. On the other hand, high levels of the branch and bound tree are not reached early on if the breadth strategy is used. Since good quality solutions are usually obtained only at high levels of the tree, the process of obtaining such solutions is also delayed. For this reason, the proposed algorithm combines the two branching strategies. Particularly, a breadth strategy is used as long as the tree level L has not reached L_1 for the first time, where L_1 is a suitable trigger parameter. The depth stragegy is implemented if $L > L_1$. With this combined strategy many candidate good quality partial assignments are formed at low tree levels. Starting with one of these solutions, the depth strategy quickly finds good quality complete solutions. If L_1 is set equal to 0, then the proposed procedure reduces to depth first, and if L_1 is set equal to m, it reduces to breadth first.

The choice of the trigger parameter L_1 is highly dependent on the dimension of the problem. A large value of L_1 increases the computer storage requirements as well as delay the attainment of good quality solutions. According to our computational experience and depending on the problem size, values of L_1 from 3 to 5 are found to be satisfactory.

2.2. Branching from an Active Node

In the proposed branch and bound procedure, the single assignment rule is used for branching from a selected active node. Particularly, an object $r\varepsilon\bar{I}$ is selected and $|\bar{J}|$ branches each corresponding to $x_{rs} = 1$ for $s\varepsilon\bar{J}$ are formed. As described previously, by using the alternative costs, some of these branches may be fathomed immediately. Some alternative procedures for selecting the particular object r for branching are given below.

2.2.1. Select object r using alternative costs

Alternative costs can be used in the process of selecting the branching object. By the use of alternative costs, it is possible to estimate the rate of increase of lower bounds associated with each object $i\varepsilon\bar{I}$. An object r in $\bar{I}$ is selected according to one of the following two rules:

1) Maximum total alternative cost rule

Choose $r\varepsilon\bar{I}$ satisfying

$$\sum_{j\varepsilon\bar{J}} \gamma_{rj} = \underset{i\varepsilon\bar{I}}{\text{maximum}} \sum_{j\varepsilon\bar{J}} \gamma_{ij}$$

Here, object r that results in the maximum sum of all lower bounds at the next tree level is selected.

2) Minimum number of branches rule

Choose $r\varepsilon\bar{I}$ satisfying

$$\sum_{j\varepsilon\bar{J}} \delta_{rs} = \underset{i\varepsilon\bar{I}}{\text{minimum}} \sum_{j\varepsilon\bar{J}} \delta_{rj}$$

where,

$$\delta_{ij} = \begin{cases} 0 \text{ if } Z_p + \gamma_{ij} \geq \mu^* \\ 1 \text{ otherwise} \end{cases}$$

Here, object r that results in the minimum number of branches at the next tree level is selected.

The proposed procedure uses a combination of the above rules. First a branching object is attempted using rule (2). If $\sum_{j\varepsilon J} \delta_{rj} < |\bar{J}|$, then object r is selected. Otherwise object r is selected using rule (1).

2.2.2. Select object r using a predetermined order

Here objects are ranked with respect to a certain criterion and object $r\varepsilon\bar{I}$ with the highest rank is selected. The following are some possible criteria for ranking the objects:

1) Maximum total interaction with all objects.
2) Maximum total interaction with unassigned objects.
3) Maximum total interaction with assigned objects.

Our computational experience suggests that selecting objects using alternative costs is superior.

2.3. Elimination of Mirror Image Branches

Two assignments π_1 and π_2 are mirror images in a quadratic assignment problem if the following hold:

1) $\pi_1 \neq \pi_2$
2) $d_{\pi_1(i)\pi_1(k)} = d_{\pi_2(i)\pi_2(k)}$ for all $i,k=1,\ldots,m;i\neq k$

In a quadratic assignment problem where the distance matrix corresponds to a rectangular layout, it is possible to identify several mirror images of a given assignment of objects. A mirror image of an assignment can be obtained by rotating the objects column-wise or row-wise such that all pairwise distances among objects remain unchanged. Hence both assignments have the same quadratic objective function value. An example of obtaining mirror image assignments is given for a 2x4 layout in Figure 1.

The above property holds for certain partial assignments also. Two partial assignments $P_1 = (I_1, J_1)$ and $P_2 = (I_2, J_2)$ are mirror images of each other if the following hold:

1) $I_1 = I_2 = I$

2) $d_{\sigma_1(i)\sigma_1(k)} = d_{\sigma_2(i)\sigma_2(k)}$ for all $i,k \varepsilon I$ $i \neq k$

3) For every $j \varepsilon \bar{J}$, there exists a location $\ell \varepsilon \bar{J}_2$ where:

   (i) $d_{j\sigma_1(k)} = d_{\ell\sigma_2(k)}$ for all $k \varepsilon I$

   (ii) $\bar{d}_1(j) = \bar{d}_2(\ell)$

Condition (1) assures that both partial solutions involve the same set of objects. The second condition asserts that all pairwise distances among assigned objects are equal in both partial assignments. The last condition states that for an unoccupied location $j \varepsilon \bar{J}$, there exists another unoccupied location $\ell \varepsilon \bar{J}_2$ such that the respective distances to locations of assigned objects and to vacant locations are equal. Obviously, if conditions (1)-(3) are satisfied then the respective lower bounds Z_{P_1} and Z_{P_2} will be equal also. Furthermore, all higher level branches emerging from P_1 and P_2 will also

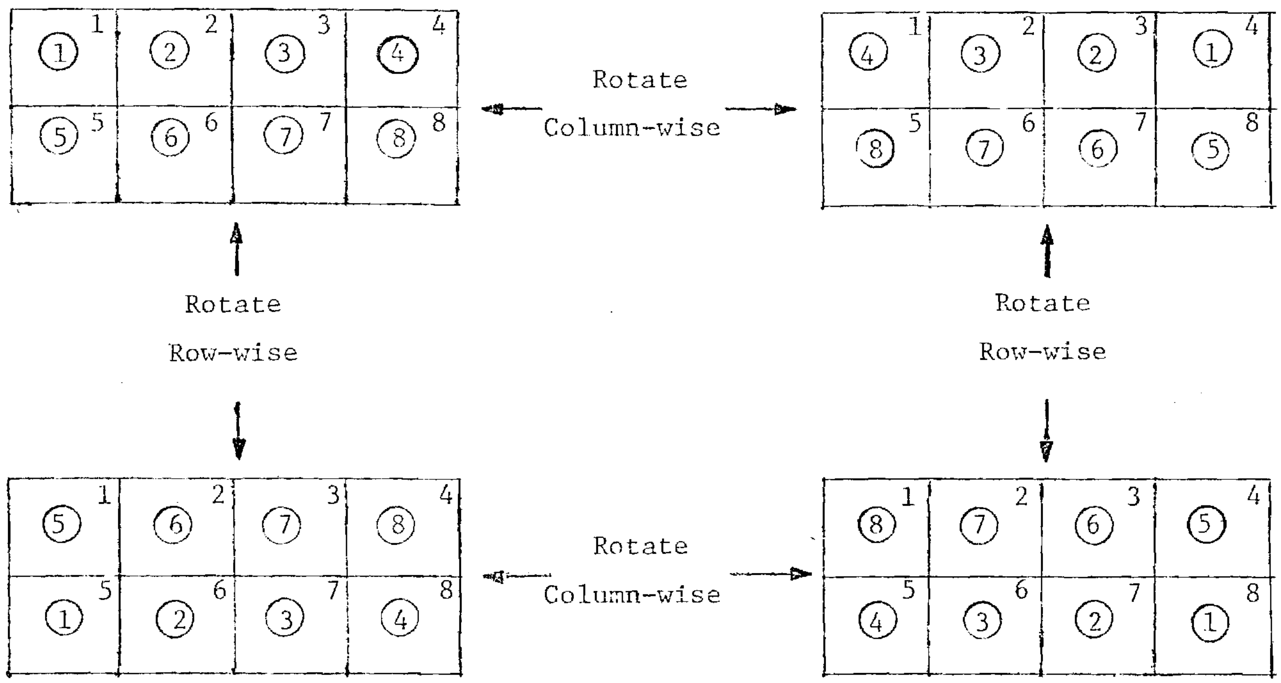

Figure 1. Mirror images of an assignment on a 2x4 layout

be mirror images of each other. Thus whenever two mirror image partial assignments are observed, one of the associated branches can be fathomed immediately.

A substantial reduction in the number of partial assignments can be achieved using the mirror image property. For example, in the 12 facility problem of Nugent et al (1968), at level 1, instead of forming 12 branches it is sufficient to create only 4 branches, resulting in a reduction of 66% in the computational effort.

The above branch and bound procedure, termed EXBB, is coded in Fortran IV. EXBB is applied to some standard quadratic assignment problems in the literature and the computational results are summarized in Table 1. In Table 2 these results are compared with other branch and bound procedures. A considerable amount of reduction in the total search effort (number of nodes and number of LAP's) is achieved with EXBB for the test problems. Also note that for QAP7 where the locations permit no mirror image assignments, the difference between the performance of EXBB and the other two methods is not clear.

Table 1. Computational Results of EXBB

Problem		Dimension m	Number of Nodes (1)	Number of LAP's solved	Optimal Objective Value	Time Cpu Sec (2)
Nugent, Vollmann and Ruml Problems [20]	QAP5	5	3	3	25	.023
	QAP6	6	5	5	43	.048
	QAP7	7	58	36	74	.372
	QAP8	8	39	28	107	.466
	QAP12	12	3385	2201	289	78.220
	QAP15	15	16001	12269	575	500.00 (3)
Elshafei's Problem (7)	QAP19	19	767	715	8606274	109.027

(1) Using branching object selection strategy alternative costs as described in Section 2.2.1

(2) On a CDC Cyber 70 Model 74-28/CDC 6400

(3) Terminated at that time without verifying optimality

Table 2. Comparison of EXBB with some other Branch and Bound Procedures

Problem	EXBB		Burkard and Stratmann (1978)		Bazaraa and Elshafei (1979)	
	Number of Nodes	Number of LAP's Solved	Number of Nodes	Number of LAP's Solved	Number of Nodes	Number of LAP's Solved
QAP5	3	3	Not Available	8	20	14
QAP6	5	5	Not Available	25	67	36
QAP7	58	36	Not Available	28	73	40
QAP8	39	28	Not Available	189	235	141
QAP12	3385	2201	Not Available	15575	37531	26368

3. AN INEXACT METHOD BASED ON BRANCH AND BOUND

When the exact branch and bound procedure is applied to the test problems in the last section, it is observed that the optimum solutions are reached early on in the search procedure. As shown in Table 3, the remaining effort is spent to prove optimality of the solution.

Table 3. Effort Spent to Prove Optimality

Problem	Total number of nodes	Node # at which optimal solution is found	% effort spent in proving optimality
QAP7	58	6	89
QAP8	39	13	66
QAP12	3385	535	84
QAP19	767	78	89

In this section, several improvement routines and methods of eliminating certain branches which are likely not to produce good quality solutions are discussed. With these modifications, the branch and bound scheme is transformed into a heuristic that produces good quality solutions within a reasonable computational effort. The major revisions to EXBB are:

1) In order to improve the quality of upper bounds, exchange routines are applied to the LAP solutions at certain branches in the search tree.

2) Since it is not possible to exhaust the search tree for large problems, several heuristics are developed for discontinuing the search at

branches where improvements are not likely even if the lower bounds indicate that fathoming is not yet achieved.

3.1. Application of the Exchange Routine

An attempt to improve the quality of the upper bound is made by applying an exchange routine to some of the LAP solutions obtained in the lower bounding process. The application of the improvement routine to all LAP solutions is not advisable. Especially at low levels of the tree, the quality of the LAP solutions is not good, so that even with the exchange routine it is usually not possible to update the upper bound.

Two levels n_1 and n_2 where $n_1 \leq n_2$ are selected. These parameters are used to trigger the exchange algorithm as follows:

1) At levels $L \leq n_1$ the exchange routine is applied to all LAP solutions. Even though at this stage it is not likely to obtain good quality solutions, the exchange routine is implemented in order to improve the solutions for use in conjunction with the variable upper bounds (Section 3.5).

2) At levels $n_1 < L < n_2$, the routine is applied only at the branch that has the least objective function value μ_p among all the branches at that level.

3) At levels $L \geq n_2$, the routine is applied at all branches in the hope of improving the quality of solutions at hand.

A suitable choice of the parameter n_1 is 1 and a good value of n_2 is around m/2.

3.2. The Exchange Routine

Most of the improvement algorithms available in the literature utilize 2-way exchanges with different implementation strategies. Some routines also use the higher order 3-way and 4-way exchanges. Los (1978) and Burkard and Stratmann (1978) discussed some procedures which employ higher order exchanges and conclude that the extra effort spent is not worthwhile.

In general, 4-way exchanges are computationally very expensive. Given an assignment of four objects (i,k,p,q), there are 23 different additional permutations. Mirchandani and Obata (1979) showed that out of these 23 permutations, 6 can be obtained by 2-way exchanges and 8 can be obtained by 3-way exchanges. Thus, only 9 permutations require 4-way exchanges. Three of these remaining 9 permutations can be easily computed by making use of 2-way exchanges.

The simultaneous exchange of locations of two pairs of objects is called a 2x2-way exchange. Specifically, consider two pairs of objects (i,k) and (p,q). Let $\Delta(i,k)$ and $\Delta(p,q)$ be the change in the objective function value for the 2-way exchanges of these two pairs of objects, respectively. Then the net change $\Delta(i,k,p,q)$ in the objective function value resulting from exchanging the locations of objects i and k and those of p and q is given below:

$$\begin{aligned}\Delta(i,k,p,q) &= \Delta(i,k) + \Delta(p,q) \\ &\quad + (f_{ip} + f_{kq} - f_{kp} - f_{iq})(d_{a(k)a(p)} + d_{a(i)a(q)} - d_{a(i)a(p)} - d_{a(k)a(q)})\end{aligned}$$

where a(i) is the location of object i in the current assignment.

The exchange routine implemented in this study evaluates 2-way and 2x2-way exchanges in the following way:

1) First improvement rule is adopted. That is, the first exchange that yields an improvement is implemented.

2) An exchange of two objects is considered only if the distance between their respective locations does not exceed a certain parameter λ.

3) The objects are ranked according to total interactions, and exchanges are performed starting with objects at the top of the list.

Using the above rules, 2-way exchanges are first performed until no improvements can be obtained. Then 2x2-way exchanges are evaluated. If any 2x2-way exchange results in a smaller objective value, the routine is reinitiated using 2-way exchanges starting from the improved solution. This procedure is terminated when no improvements are possible using the 2x2-way exchange routine.

3.3. Selective Location Rule

In optimal or good quality solutions it is generally expected that objects with large total interaction are assigned to "median" locations while objects with small total interaction are assigned to "off-median" or corner locations. While branching from a partial assignment, new branches are created by assigning the selected object to each of the vacant locations. The assignment of a high interaction object to a corner location is likely not to lead to a good quality solution, even if the lower bound does not exceed the incumbent objective value. In general, a substantial computational effort may be expended in pursuing such "bad" branches. By the selective location rule, high interaction objects are assigned only to "cental" locations and low interaction objects are assigned to "off-median" locations. Obviously, there is no guarantee that all optimal or good quality solutions must satisfy these additional restrictions, but it is hoped that the exchange routine would help overcome this difficulty.

The objects and locations are ranked according to non-increasing total interactions and non-decreasing total distances, respectively. A parameter t is chosen and for each object i, the set of permissible locations T_i is determined as follows. Let i^* be the rank of the object i. Then:

$$T_i = \{J: i^* - t \leq j^* \leq i^* + t\}$$

where j^* is the rank of location j. A good choice of t is in the range from $^m/3$ to $^m/2$.

3.4. Group Assignment of Objects

In an optimal solution we would generally expect that the set of objects with large pairwise interactions to be located close to each other. This observation is incorporated in the branch and bound procedure as follows. Choose a parameter ε denoting the threshold percentage of total interactions. Suppose that at branch P object r is assigned to location s, and that the solution of LAP yields the complete assignment π_p. Then for all $i \varepsilon \bar{I}$ satisfying:

$$\frac{f_{ri}}{\sum_{k=1}^{m} f_{rk}} \geq \varepsilon$$

object i is assigned to the location $\pi_p(i)$ for all completions of the current partial assignment P. That is, P is updated as follows:

$$P \leftarrow P \cup \{(i,\pi_P(i)\}$$

By this procedure it is hoped to speed the branch and bound scheme and at the same time generate good partial assignments. A suitable choice of ε is around .25.

3.5. Variable Upper Bounds

In order to reduce the search effort, variable upper bounds are used. Bazaraa and Elshafei (1977) discussed fictitious upper bounding procedures for tree search algorithms and applied them to the quadratic assignment problem in [2]. Also, Burkard and Stratmann (1978) implemented a variable upper bounding scheme to an inexact branch and bound procedure for the same problem. The concept of fictitious or variable upper bounding can be explained as follows.

A branch is fathomed if its lower bound is greater than or equal to the incumbent upper bound. But in quadratic assignment problems, the upper bound usually exceeds the lower bound except for large tree levels. Thus a substantial amount of effort is typically spent is pursuing bad quality solutions before fathoming can be achieved. In order to speed fathoming, a fictitious upper bound $V(L) \leq \mu^*$ is set for each level L of the search tree. The branches at level L that have a larger lower bound than V(L) are fathomed. These fictitious upper bounds are called variable upper bounds since their values depend on the tree level L, where $V(L) \leq V(L + 1)$ for $L=1,\cdots,m-1$. The variable upper bounds must be such that good quality branches are not fathomed. Only the branches which are likely not to produce improved solutions are fathomed.

In order to determine the form of the variable upper bounds, the gap between the lower bounds and the best known solutions for some test problems are examined. As Burkard and Stratmann (1978) suggested, the gap decreases quadratically with respect to the level of the tree. A proper function for the variable upper bounds V(L) is of the form:

$$V(L) = \begin{cases} Z_0 + g_1 L + g_2 L^2 & L=1,\ldots,\theta \\ \mu^* & L=0 \text{ or } L>\theta \end{cases}$$

where θ is a suitable parameter corresponding to the tree level at which lower and upper bounds are usually equal. The values of coefficients g_1 and g_2 are determined by the following boundary conditions:

$$V(\theta) = \mu^*$$

$$2V(\theta/2) - Z_0 = \mu^*$$

The computational experience with variable upper bounds show that relatively good quality solutions are obtained early on and further efforts either do not improve the current upper bound or produce little improvement. Since for a fixed L, an increased θ results in decreasing V(L) and hence speeding the fathoming process, the parameter θ is incremented by one after evaluating certain number of nodes. For Gilmore-Lawler bounds, a good starting value for θ is around $\frac{2}{3}m$. Furthermore, the branch and bound procedure is terminated if no improvements over the current upper bound are obtained within a specified amount of time.

The exact branch and bound procedure EXBB of Section 2 is modified using the above heuristic strategies, resulting in the code INBB. A variant of this inexact procedure that differs only in the computation of lower bounds is also developed. INRO uses Roucairol's reduction procedure [22]. The basic idea of Roucairol's procedure is to reduce the interaction and distance matrices so that the quadratic assignment problem is written as:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} f'_{ik} d'_{j\ell} x_{ij} x_{k\ell} + \sum_{i=1}^{m} \sum_{k=1}^{m} c_{ij} x_{ij} + \nu_0$$

$$\text{Subject to:} \quad x \varepsilon X$$

where,

f'_{ik} = nonnegative reduced flow from object i to object k

$d'_{j\ell}$ = nonnegative reduced distance between locations j and ℓ

c_{ij} = reduced cost of locating object i to location j

ν_0 = a fixed cost obtained by the reduction process.

Ignoring the reduced quadratic part, the lower bound Z_P for a given partial assignment P is computed by solving the following LAP:

$$Z_P = \underset{x \varepsilon X_P}{\text{Minimum}} \sum_{i \varepsilon \bar{I}} \sum_{j \varepsilon \bar{J}} h_{ij} x_{ij} + \nu_P + \nu_0$$

where;

$$h_{ij} = c_{ij} + \sum_{k \varepsilon I} f'_{ik} d'_{j\sigma(k)}$$

$$\nu_P = \sum_{i \varepsilon I} c_{i\sigma(i)} + \sum_{i \varepsilon I} \sum_{k \varepsilon I} f'_{ik} d'_{\sigma(i)\sigma(k)}$$

Obviously, the above lower bound can be strengthened by computing a lower bound on the reduced quadratic term. But our experience with the test problems suggest that including the reduced quadratic term into the lower bound computations does not necessarily yield stronger lower bounds in comparison with Gilmore-Lawler procedure.† Furthermore as far as the total effort is concerned, by including the reduced quadratic relations into the bound computations, the computational advantage of Roucairol's procedure over the Gilmore-Lawler method is lost.

The modified branch and bound schemes INBB and INRO are applied to the problems of Nugent et al (1968). For each problem, the following three strategies for selecting the branching object are used.

1) Maximum total interaction with all objects

2) Maximum total interaction with already assigned objects

3) The alternative cost rule described in Section 2.2.1.

The computational results are given in Table 4. As seen from the table, even with the weaker bounding procedure INRO, good quality solutions are obtained in a relatively small amount of computational time. Furthermore, it is observed that the selection rule of the branching object does not affect the quality of the solutions significantly. An attempt to further improve the quality of the solutions at hand is made using a routine that implements 2, 3, and 4-way exchanges. This routine is similar to the Mixed Exchange Algorithm of Mirchandani and Obata (1979). Either very little improvement or none at all

---

†For problems that have a rectangular grid layout and an interaction matrix with at least one zero in each row and column, it can be shown that Roucairol's procedure cannot yield stronger lower bounds even if the reduced quadratic term is included in the bound computations.

Table 4. Summary of the Results for Nugent et al (1968) Problems

Problem	Dimension m	Branching Object Selection Strategy	INRO		INBB		Best known value in literature [3]
			Value	Cpu. Sec. (a)	Value	Cpu. Sec. (a)	
QAP15	15	(1)	576	11.85	575	31.49	575
		(2)	576	16.55	575	32.74	
		(3)	575	10.80	575	30.59	
QAP20	20	(1)	1297	30.75	1285	100.00	1285
		(2)	1285	29.50	1285	100.00	
		(3)	1300	50.00	1285	100.00	
QAP30	30	(1)	3079	176.19	3079	400.83	3077
		(2)	3083	139.49	3080	346.95	
		(3)	3080	200.00	3078	400.84	

(a) On a CDC cyber 70 model 74-28/CDC 6400 is attained as a result of these computationally expensive exchange routines, rendering their use unjustified.

4. AN ITERATIVE APPLICATION OF THE INEXACT BRANCH AND BOUND PROCEDURE

One disadvantage of the proposed inexact methods of Section 3 is that their effectiveness is highly dependent on the quality of the initial partial assignments selected at low tree levels. Since the search tree is not exhausted, these methods commit themselves to the partial solutions selected at low levels. In order to reduce this dependency on initial partial assignments the following iterative branch and bound procedure is developed.

The procedure applies the inexact branch and bound scheme iteratively by alternating between two branching rules. At each iteration, several number of objects are fixed at the locations of the previous iteration and the branch and bound computations are performed by changing the branching rule. The procedure terminates when no improvements are obtained, and is summarized as follows.

Form two ordered sets of objects S and $\bar{S}$ such that:

$$\sum_{k=1}^{m} f_{i_\ell,k} \geq \sum_{k=1}^{m} f_{i_{\ell+1},k} \qquad \text{for all } i_\ell \varepsilon S \qquad \ell=1,\ldots,m$$

and

$$\sum_{k=1}^{m} f_{i_\ell,k} \leq \sum_{k=1}^{m} f_{i_{\ell+1},k} \qquad \text{for all } i_\ell \varepsilon \bar{S} \qquad \ell=1,\ldots,m$$

Let s_q be a parameter corresponding to the branching object selection rule at iteration q, where:

$$s_q = \begin{cases} 1 & \text{select the branching object having maximum total interaction with all objects.} \\ 2 & \text{select the branching object having minimum total interaction with all objects.} \end{cases}$$

Step 0: Set $q=1$, $s_q=2$, $\bar{s}_q=1$, $\mu^*=\infty$, and $\pi^*=0$.

Let $P_0 = (I_0,J_0) = (\emptyset,\emptyset)$ and go to Step 1.

Step 1: Apply INBB using the branching rule s_q, and starting with the partial solution P_0. Terminate the branch and bound procedure whenever the bottom of the search tree is reached for the first time. Let the best solution found be μ_q and π_q and go to Step 2.

Step 2: If $\mu_q > \mu^*$ or $\mu_q = \mu_{q-2}$ stop. Otherwise go to Step 3.

Step 3: Let $\mu^* = \mu_q$ and $\pi^* = \pi_q$. Determine a number k_q which corresponds to the number of objects to be fixed at this iteration. Set $P_0 = (I_0,J_0)$ as follows:

$$I_0 = \begin{cases} \text{first } k_q \text{ elements of ordered set } S & \text{if } \bar{s}_q=1 \\ \text{first } k_q \text{ elements of ordered set } \bar{S} & \text{if } \bar{s}_q=2 \end{cases}$$

and

$$J_0 = \{\pi^*(i): \ i \varepsilon I_0\}$$

Let,

$$s_{q+1} \leftarrow \bar{s}_q$$

$$\bar{s}_{q+1} \leftarrow s_q$$

$$q \leftarrow q+1$$

and return to Step 1.

The procedure terminates at an iteration q if the objective value of the assignment found at that iteration is greater than the incumbent μ^* or equal to that obtained by the previous iteration which has the same branching rule. The number of objects k_q to be fixed at each iteration is around m/3.

The above iterative procedure is applied to Nugent et al (1968) and Steinberg (1961) problems and the results are summarized in Table 5. The procedure produced at the best known solutions for problems QAP20 and QAP34-1. For Problems QAP30 and QAP34-2 the iterative procedure produced solutions which are better than the best known in the literature. These solutions are given in the Appendix.

Table 5. Summary of Iterations for Nugent et al (1968) and Steinberg (1961) Problems with Iterative Branch and Bound Scheme

	Problem	Dimension m	Iteration q	Branching Order s_q	Number of fixed objects k_{q-1}	Objective value	Time Cpu. Sec. (a)	Cumulative Computation Time Cpu. Sec. (a)	Best known in literatur [3]
Nugent, Vollman and Ruml (1968) Problems	QAP20	20	1	2	0	1303	52.62	52.62	1285
			2	1	6	1298	28.32	70.94	
			3	2	6	1298	19.24	90.18	
			4	1	8	1292	20.37	110.55	
			5	2	8	1285	20.08	130.63	
			6	1	10	1285	15.40	156.03	
	QAP30	30	1	2	0	3130	130.53	130.53	3077
			2	1	8	3072	61.91	192.44	
			3	2	8	3064	67.11	259.55	
			4	1	12	3064	31.20	290.75	
			5	2	12	3064	29.50	320.25	
Steinberg (1961) Problems	QAP34-2 Squared Eucli-dean Distance	34	1	2	0	7999	398.22	398.22	7926
			2	1	13	7926	46.56	444.78	
			3	2	16	7926	62.26	506.71	
	QAP34-2 Rectin-linear Distance	34	1	2	0	4802	326.60	326.60	4802
			2	1	13	4800	55.87	382.47	
			3	2	16	4800	83.26	465.73	

(a) On CDC Cyber 70 model 74-28/CDC 6400

APPENDIX

SOLUTIONS FOR QAP30 and QAP34-2

Problem: QAP30

Best found Value: 3064

Assignment:

1	2	3	4	5	6
15	23	11	30	4	14
7	8	9	10	11	12
17	18	8	16	27	20
13	14	15	16	17	18
1	22	7	19	3	29
19	20	21	22	23	24
24	26	10	9	21	2
25	26	27	28	29	30
12	6	25	13	28	5

Problem: QAP34-2

Best found Value: 4800

Assignment:

1	2	3	4	5	6	7	8	9
26	22	27	23	11	6	5	3	
10	11	12	13	14	15	16	17	18
25	21	14	20	12	13	4	8	2
19	20	21	22	23	24	25	26	27
24	32	19	28	1	7	10	18	17
28	29	30	31	32	33	34	35	36
33	34	31	30	29	15	9	16	

REFERENCES

[1] Bazaraa, M. S. and A. N. Elshafei, "On the Use of Fictitious Bounds in Tree Search Algorithms," Management Science 23, 904-908 (1977).

[2] Bazaraa, M. S. and A. N. Elshafei, "An Exact Branch and Bound Procedure for Quadratic Assignment Problems" Naval Research Logistics Quarterly 26, 109-121 (1979).

[3] Bazaraa, M. S. and H. D. Sherali, "Benders' Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem," Naval Research Logistics Quarterly 27, 29-41 (1980).

[4] Burkard, R. E. and K. H. Stratmann, "Numerical Investigations on the Quadratic Assignment Problem," Naval Research Logistics Quarterly 25, 129-148 (1978).

[5] Christofides, N. A. Mimgozzi, and P. Toth," Contributions to the Quadratic Assignment Problem," European Journal of Operational Research 4, 243-247 (1980).

[6] Edwards, C. S., "A Branch and Bound Algorithm for the Koopmans-Beckman Quadratic Assignment Problem," Mathematical Programming Study 13, 35-52 (1980).

[7] Elshafei, A. N., "Hospital Layout as a Quadratic Assignment Problem," Operational Research Quarterly 28, 167-179 (1977).

[8] Frieze, A. M. and J. Yadegar, "On the Quadratic Assignment Problem," Queen Mary College, March (1981).

[9] Gavett, J. W. and N. V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," Operations Research 14, 210-232 (1966).

[10] Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," SIAM Journal on Applied Mathematics 10, 305-313 (1962).

[11] Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Management Science 17, 453-471 (1970).

[12] Kaufman, L. and F. Broeckx, "An Algorithm for the Quadratic Assignment Problem Using Benders' Decomposition," European Journal of Operational Research 2, 204-211 (1978).

[13] Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica 25, 53-76 (1957).

[14] Land, A. M., "A Problem of Assignment with Interrelated Costs," Operational Research Quarterly 14, 185-198 (1963).

[15] Lawler, E. L., "The Quadratic Assignment Problem," _Management Science_ 19, 586-590 (1963).

[16] Little, J. D. C., K. G. Murty, D. W. Sweeney and K. Caroline, "An Algorithm for the Travelling Salesman Problem," _Operations Research_ 14, 972-989 (1963).

[17] Los, M., "Comparison of Several Heuristic Algorithms to Solve Quadratic Assignment Problems of the Koopmans-Beckmann type," Presented at the _International Symposium on Locational Decisions_, Held at Banff, Alberta, April (1978).

[18] Love, R. F. and J. Y. Wange, "Solving Quadratic Assignment Problems with Rectangular Distances and Integer Programming," _Naval Research Logistics Quarterly_ 23 623-627 (1976).

[19] Mirchandani P. and T. Obata, "Algorithms for a Class of Quadratic Assignment Problems," Presented at the _Joint National TIMS/ORSA Meeting_, New Orleans, May (1979).

[20] Nugent, C. E., T. E. Vollmann, and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," _Operations Research_ 16, 150-173 (1968).

[21] Pierce, J. F. and W. B. Crowston, "Tree Search Algorithms for Quadratic Assignment Problems," _Naval Research Logistics Quarterly_ 18, 1-36 (1971).

[22] Roucairol, C., "A Reduction Method for Quadratic Assignement Problems," _III Symposium Uber Operations Research_, Universität Mannheim, October (1978).

[23] Sherali, H. D., "The Quadratic Assignment Problem: Exact and Heursitic Methods," Unpublished Doctoral Dissertation, Georgia Institute of Technology, June (1979).

[24] Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," _SIAM Review_ 3, 37-50 (1961).