EXTENSIONS OF A ROUTE STRUCTURING ALGORITHM

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

James Oscar Bell

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Operations Research

Georgia Institute of Technology

October, 1974

EXTENSIONS OF A ROUTE STRUCTURING ALGORITHM

Approved:

Gunter P. Sharp, Chairman

P. S. Jones

D. B. Young

Date approved by Chairman: 23 Sep 74

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation for the valuable guidance, criticism, encouragement, and patience given by Dr. Gunter P. Sharp who acted as thesis advisor.

Special thanks also go to Dr. Donovan B. Young and Dr. Paul S. Jones, who served on his thesis committee and provided many constructive comments and helpful recommendations.

Dr. Robert N. Lehrer and Dr. William W. Hines deserve my appreciation for providing financial support, and again appreciation to Dr. Donovan B. Young who acted as my assistantship advisor.

Finally, a special debt is owed to Jent and my many friends in the Industrial and Systems Engineering School and at Georgia Institute of Technology for bearing with me through my graduate work.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

SUMMARY

In previous work Sharp developed an heuristic procedure for solving in a unified way the transit network design problem that follows after estimating the travel demand and determining the station locations: How to simultaneously select fixed-cost guideways for inclusion in the network, determine vehicle routes and route service frequencies, and assign passengers to origin-destination paths in order to minimize the total of construction costs, passenger travel and delay time costs, and vehicle operating costs, while satisfying total transportation demand.

This research extends the above work in two main areas. The first is restructuring of vehicle routes to provide better matching of service to demand, reduce passenger transfers, and reduce duplication among route sections. The second is the allowance of multiple guideway changes, as opposed to one-at-a-time insertion or deletion.

Sharp's FORTRAN V computer program was modified and extended by adding subroutines for route restructuring and by changing the switching rules controlling the guideway insertions and deletions. These changes resulted in fewer vehicle routes and lower total costs, the latter due mainly to a reduction in passenger travel time costs. The solution procedure as modified obtained an average 3% reduction in total costs with an average 3% increase in computation time. The largest problem contained twelve stations, twenty-five possible two-way guideways, fifty commodities, and a maximum of fifteen vehicle routes. The computation time ranged from 1.1 to 11.4 minutes on a UNIVAC-1108.

CHAPTER I

INTRODUCTION

Background

During the current energy crisis a need is being felt for good public rapid transit. Presently very few of the rapid transit systems that exist in the United States can be described as efficient. Bay Area Rapid Transit (BART) in San Francisco is being operated manually, and not by automatic controls as designed. The Huntington, West Virginia, system, an ideal situation for mass rapid transit, is in the process of being abandoned and possibly torn down. The Metropolitan Atlanta Rapid Transit Authority (MARTA) is experiencing difficulties in its early stages.

The reasons for these failures and setbacks are many. A few are:

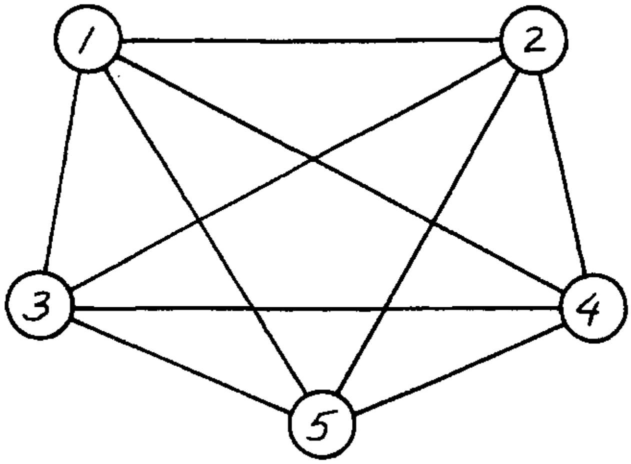
1) Inefficient route scheduling. Passengers, the major source of income for most systems, are not satisfied unless the vehicle takes them from origin to destination with what they perceive as a minimum of delay, i.e. travel times and transfer times that do not admit of obvious improvement.

2) Inadequate frequency of vehicle service. More frequent service is the passenger's desire, however costly to the transit company. Too little frequency of service increases the cost efficiency of the system but also increases passenger waiting time and hence makes the system less desirable. As a result, travel demand declines, leading to reduced revenues and eventual further reductions in service frequency.

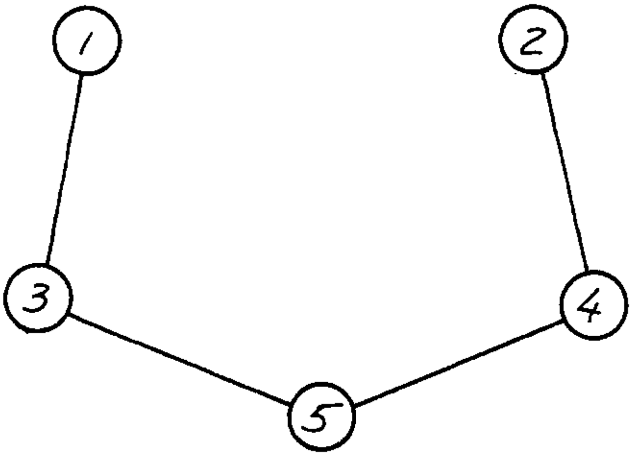
3) Guideway consideration. A construction-oriented guideway system, as shown in Figure 1, is a minimally spanned network that connects all of the stations. This is usually efficient only in that it has low construction costs. Offsetting this advantage is the fact that passenger travel time is relatively high. A passenger-oriented guideway system is a maximal spanning network, allowing the passenger to get from his origin to his destination in a minimal travel time. This, however, increases construction costs considerably.

4) Cost overruns. The final downfall of most transit systems is the result that summarizes all of their problems--cost. The more problems a system has, the more it will spend trying to solve or correct them. If it cannot acquire funding from one of the government agencies it is forced to raise fares, which results in reduced clientele.

To date, much work has been accomplished on aspects of designing an efficient transit system, from estimating highway planning needs by the Highway Planning Package (14) or public transit needs by the Urban Mass Transit Authority System (16), to theoretical solutions for multi-commodity transshipment and network flow models. Most work in this area has been on a sequential basis. The planning procedure usually follows a sequence of steps: trip generation, trip distribution, station location, guideway location, trip assignment, vehicle routing, and vehicle scheduling. The irreversibility of decisions made early in this sequential procedure inevitably biases the final result and reduces the chances of achieving optimal or near-optimal solutions. To overcome the disadvantages of the sequential planning process a number of researchers have developed solution procedures that involve the simultaneous consideration of guideway construc-



Traveler-Oriented Design: Maximally Connected Network



Construction-Oriented Design: Minimal Spanning Tree

Figure 1. Comparison of Traveler-Oriented Design to Construction-Oriented Design

tion costs and passenger travel time, such as Scott (12), Billheimer (4), and Sharp (13). In particular, Sharp developed a solution procedure for simultaneously selecting fixed-cost guideways, determining vehicle routes and route service frequencies, and assigning passengers to origin-destination paths in order to minimize the total of construction costs, passenger travel and delay time costs, and vehicle operating costs while satisfying total demand. This solution procedure uses techniques based on arc insertion-deletion routines developed by Scott and Billheimer.

Purpose

Sharp's solution procedure is not ready completely to solve problems of a realistically sized transit system; however, it can be used to evaluate and compare solutions to these problems and to give a general direction in which to proceed. The present research was done to attempt to improve the solution procedure, not necessarily to reduce execution time, but especially to improve the quality of solutions obtained. The work was concentrated in two main areas, route structuring and guideway insertion-deletion, these being of critical importance in the design of any transit system.

Due to the complexity of the problem, the route structuring algorithm and the guideway insertion-deletion algorithm will be presented only in general form here, with a more thorough explanation given after Sharp's solution procedure has been discussed.

Route Restructuring

Sharp's procedure does not attempt to restructure existing routes. The only way a route can be increased in length is if an arc is appended to it. Duplication of routes is also frequent due to route construction, or guideway deletion which deletes parts of routes and may reduce two different routes to similar ones. The work done in the present research on route restructuring was limited to adding the following features not included in Sharp's procedure: combining two routes at a common node, reversing routes, or deleting a route if it already exists elsewhere in the system. These manipulations are made if costs resulting from vehicle and passenger costs are decreased.

Multiple Guideway Changes

Sharp's guideway algorithm progresses on a sequential basis by inserting or deleting guideways singly if costs are thereby decreased. However, due to the nature of the problem, a need arises to allow for temporary cost increases if an overall cost decrease can be achieved later. This is the idea behind multiple guideway changes--an arc is allowed to be inserted at a cost increase if and only if an arc can then be deleted with a net cost decrease. The work done in the present research on multiple guideway changes focused on ways to replace a given set of guideways with a lower-cost set without restricting such replacements only to those in which every insertion or deletion decreases overall costs by itself.

CHAPTER II

PREVIOUS WORK

The problem of simultaneously determining guideways, trip assignments, and vehicle routes and schedules, when stated in mathematical terms, becomes a very large and complex mathematical programming problem, even for a small transit system. To date, subproblems have been examined and small problems solved. However, not much work has been accomplished in solving problems of realistic size. Since this research focuses on route scheduling and on guideway insertion-deletion, the literature search will review those areas.

Route Scheduling

Much of the work done in route scheduling is in the area of minimizing fleet size subject either to a fixed schedule of trips between stations or to fixed demands between stations. Because of the complexity of route scheduling, due to the number of possible combinations arising even in small problems, the work has been limited in this area.

Young (18) formulated a method for optimizing a vehicle route schedule based on a recursive programming algorithm that computes a "currently optimal" schedule for a single vehicle. Each successive vehicle schedule optimization is designed to improve the quality of the overall fleet schedule. After all vehicles are initially assigned, the vehicles are singly rechecked to see if any can be rescheduled to improve total quality. The function to be maximized consists of three basic components: operating costs, revenues, and traveler benefits.

Hyman and Gordon (8) also used a dynamic programming approach to formulate a solution procedure to attract the maximum number of passengers under time-varying demands in commercial airline scheduling. The objective was to develop a good schedule such that an optimal load factor-frequency combination would maximize earnings within constraints. The model used determines an initial vehicle assignment and iteratively obtains schedules for the remainder of the day. This is a one-pass operation; that is, once a schedule for a vehicle is set, it is not changed.

Bartlett and Charnes (2) scheduled trains to minimize the number of vehicles required to meet total demands. Their model used a linear programming approach and was limited to a small system, restricting routes to be cyclic with no more than three arcs.

Gaskell (7) examined five methods to determine a near-optimal set of routes by which vehicles at a depot would supply the customers with their respective loads. The optimal solution here is regarded as that which minimizes the number of routes, hence the number of vehicles, and, for them, the total mileage required. The first method examined was a "visual" assignment. The optimal set was obtained for each of the six cases run, but the time spent on this solution was excessive. The other four methods are based on the savings incurred by combining a route with an existing route, thereby decreasing the total mileage. Exclusive of the visual method, none of the four were uniformly better than any other and the outcome is not sensitive to the method used.

Christofides and Elion (5) examined the truck-dispatching problem by three possible methods: branch and bound, the "saving" approach, and the 3-optimal tour method. The branch and bound approach was formulated as a traveling salesman problem by eliminating the real depot and replacing it by N artificial depots, all located in the same position. The number of vehicles employed is thus N. A lower bound for N is the sum of the customer requirements divided by vehicle capacity. Before branching to a new node, vehicle capacity and total vehicle tour length are checked. The second method is based on the savings involved by linking routes together. The third method is based on observing that the minimal traveling salesman tour does not intersect itself. Starting with an arbitrary initial intersectionless tour, two links are deleted and the four end points are connected by two new links. This is repeated until no improvements can be made by eliminating two links and replacing them by two others, termed "2-optimal." This combination is then repeated for three links until the 3-optimal solution is obtained.

Guideway Insertion-Deletion

Most of the previous work done in guideway insertion-deletion concentrates on determining which fixed-cost arcs need to be open to satisfy total flow demands while minimizing an overall objective function or satisfying a constraint on total arc lengths. Scott (12) formulated two approximate algorithms to establish a set of arcs linking together a given set of nodes such that the sum of the shortest distances through the resulting network between pairs of vertices is a minimum while the total cost of the selected arcs is less than a specified upper bound. He developed two separate procedures for solving this problem, a forward algorithm to insert arcs and a backwards algorithm to delete arcs. The forward algorithm starts with a minimal spanning tree and iteratively selects an arc to insert in such a way that the greatest possible reduction in the value of the objective function is achieved without violating the upper bound of the total arc costs. The backwards algorithm starts with a maximally connected network and successively deletes arcs to produce the least increase in the value of the objective function. This continues until the total arc costs are less than the specified upper bound. Twenty-six networks were examined, containing from seven to ten nodes. Computations time on an IBM/365 was less than one minute for each problem. The forward algorithm gave twenty-one fully optimal solutions, with a departure from optimality in the five suboptimal solutions of less than 10.4%. The backwards algorithm produced twenty-four optimal solutions, with departures from optimality in the two suboptimal solutions of less than 2.3%.

The solution of the forward algorithm is dependent upon the minimal spanning tree used as a starting point. If one branch is included which should not be in the optimal solution, then the forward algorithm will never reach the optimal set since arcs cannot be deleted.

Billheimer's (4) work represents an extension of Scott's backwards algorithm. He begins with a maximally connected network but allows for both arc deletion and insertion. In the deletion phase of the algorithm, flows are routed and improvement parameters are computed based on decreased fixed costs from closing an arc and increased variable costs from rerouting flow to the second-shortest paths. When no more deletions can be made to reduce costs, similar improvement parameters are computed for inserting arcs. The process iterates between arc deletion and arc insertion until no more changes can be made to improve the solution.

Bergendahl (3) designed a solution procedure to determine roads to be constructed while accounting for decreases or increases in flow on near-by roads due to vehicles utilizing the new route. The problem is divided into time periods for successive additions to the road network. The optimal operation of the network during each time period can then be formulated as a multicommodity network flow problem. The overall best solution is then obtained through dynamic programming.

O'Connor and DeWald (11) formulated a solution procedure to determine the optimal guideway network needed to satisfy travel demands while ensuring no passenger has travel time exceeding a limit based on his direct or shortest-path travel time. They start with a maximally connected network and compute total costs. An arc is then deleted and replaced by another, and the system is rechecked. This is done for all arcs, and the feasible set with the minimal total cost is compared to the previous network (initially the maximally connected network). If total guideway and travel time costs are lowered, this new network replaces the previous network as the best, and the solution procedure is repeated. The process terminates when either no arcs can be removed to retain feasibility or the minimal feasible network is worse than the previous network. With n nodes, there are initially n(n-1) arcs; therefore, computation is very extensive even for a small system and excessive for a realistically sized system. Convergence of the algorithm was demonstrated up to a four-node problem. For the four-node problem an upper bound on the number of new networks generated is between thirteen and sixty-nine; for five nodes, twenty-one and one hundred ninety-six.

Other approaches have been developed for determining guideway sets, such as those of Ochoa-Rosso (10), Yaged (17), Ishmael (9), and Aburto-Avila (1). These efforts generally have been directed at problem formulations different from that pursued in this research.

CHAPTER III

PROBLEM STATEMENT AND OVERALL SOLUTION PROCEDURE

Specific Problem Statement

The problem of determining guideways, trip assignments, vehicle routes and route service frequencies can conveniently be represented in network terms. Consider a given connected network (N,A), with the set of nodes N representing station locations and the set of one-way arcs A representing possible one-way guideways. These station locations and possible guideways are predetermined by the transportation planner. The one-way guideways are assumed to be paired and uncapacitated.

Travel demand occurs between node pairs ij, and each passenger traveling in the system is one unit of flow. Since passengers traveling from i to j are not interchangeable with passengers traveling from k to ℓ, the units of flow must be differentiated according to origin and destination. All passengers traveling from the same origin node to the same destination node are assumed to use the same path; thus they become a single commodity. Hence, the travel demands render the problem a multicommodity transshipment problem.

A vehicle route consists of a series of connected arcs, oriented so that the head of one is incident to the tail of the next, or in network terms, a path. If the route ends at the same node as its starting point, the path is a cycle. In this problem all vehicle routes must be cycles, with vehicles stopping at all stations in a cycle.

Passenger flow units must be assigned to trips, which are paths that are contained in a combination of one or more vehicle routes. The guideways are uncapacitated, but the passenger flow assignments are limited by the capacity and frequency of service of the vehicles assigned to the routes utilized. Routes are restricted to open, or selected, guideways only.

Each possible guideway (i,j) has a fixed cost p_{ij} assigned to it, a variable passenger travel time d_{ij} which is common to all commodities using that arc, and a variable operating cost c_{ij}. It is assumed that "vehicles" throughout the system operate singly or in a single capacity combination throughout each cyclical route. Two types of passenger delay costs are considered, a waiting cost and a transfer cost. The waiting cost d_w is incurred when a passenger is boarding a route at his origin node and is inversely proportional to the frequency of service of that route on the arc to be traversed. The transfer cost d_t is incurred whenever a passenger must transfer from one vehicle route to another.

Formulation of Problem

Using the above terminology, the problem is to select for a network a subset of the set of arcs A, to assign vehicles to routes subject to feasibility constraints, and to route flow on the network subject to multicommodity flow requirements and vehicle capacity constraints on the arcs. The objective is to minimize the total of construction costs p_{ij}, passenger travel costs d_{ij}, passenger delay and transfer costs d_w and d_t, and vehicle operating costs c_{ij}. All costs are expressed in equivalent time units, and we shall use the words "cost" and "time" interchangeably.

$$\text{Minimize } Z = \sum_{(i,j)\in A} u_{ij}\, p_{ij} + \sum_{k\in K} \sum_{m\in M} \sum_{(i,j)\in A} f^k_{mij}\, d_{ij} \tag{3-1}$$

(guideway fixed costs) (travel time costs)

$$+ d_w \sum_{k\in K} \sum_{m\in M} v^k_m / y_m + d_t \sum_{k\in K} s^k$$

(waiting time costs) (transfer time costs)

$$+ \sum_{m\in M} \sum_{(i,j)\in A} y_m\, a_{mij}\, c_{ij}$$

(vehicle operating costs)

Subject to

Passenger flow requirements at each node:

$$\sum_{m\in M} \sum_{j\in A(i)} f^k_{mij} - \sum_{m\in M} \sum_{j\in B(i)} f^k_{mji} = r^k_i, \quad k\in K, i\in N \tag{3-2}$$

Vehicle capacity on each arc:

$$\sum_{k\in K} f^k_{mij} \le y_m a_{mij} g, \quad m\in M,\ (i,j)\in A \tag{3-3}$$

Arc feasibility:

$$\sum_{m\in M} y_m\, a_{min} \le u_{ij} U, \quad (i,j)\in A \tag{3-4}$$

Symmetric arc property:

$$u_{ij} - u_{ji} = 0, \qquad (i,j)\epsilon A \tag{3-5}$$

Waiting time occurrences:

$$v_m^k = \frac{1}{2} \sum_{(i,j)\epsilon A} \left| f_{mij}^k - \sum_{\ell\epsilon N} f_{mj\ell}^k \right| , \ k\epsilon K,\ m\epsilon M \tag{3-6}$$

Transfer time occurrences:

$$s^k = \frac{1}{2} \sum_{m\epsilon M} \sum_{(i,j)\epsilon A} \left| f_{mij}^k - \sum_{\ell\epsilon N} f_{mj\ell}^k \right| - 1,, \ k\epsilon K \tag{3-7}$$

Integrality and nonnegativity:

$$u_{ij} = 0,\ 1, \qquad (i,j)\epsilon A \tag{3-8}$$

$$f_{mij}^k \le 0, \qquad k\epsilon K,\ m\epsilon M,\ (i,j)\epsilon A \tag{3-9}$$

$$y_m = \text{a nonnegative integer}, \qquad m\epsilon M \tag{3-10}$$

$$v_m^k = \text{a nonnegative integer}, \qquad k\epsilon K,\ m\epsilon M \tag{3-11}$$

$$s^k = \text{a nonnegative integer}, \qquad k\epsilon K \tag{3-12}$$

where

(N,A) is the given network of nodes $i \in N$ and possible arcs $(i,j) \in A$

K set of commodities, or travel demands between node pairs ij

M set of cycles, or vehicle routes

$A(i)$ $\{j \mid (i,j) \in A\}$, or "after i"

$B(i)$ $\{j \mid (j,i) \in A\}$, or "before i"

u_{ij} incidence variable for arc (i,j), $u_{ij} = 1$ if the arc is open (meaning the guideway connecting stations i and j are constructed), $u_{ij} = 0$ otherwise

f^k_{mij} flow of commodity k on arc (i,j) in route m during the operating time period, or the number of passengers of commodity k traveling from i to j on a vehicle on route m

y_m number of vehicles traversing route m per operating time period, with vehicles making stops at each station in the route

v^k_m number of waiting time occurrences of commodity k on route m during the operating time period, or the number of passengers of commodity k boarding route m

s^k number of transfer occurrences of commodity k from one route to another during the operating time period, or the number of passengers of commodity k undergoing a transfer as part of their origin-destination journey

p_{ij} fixed cost of opening arc (i,j), or the cost of opening the guideway connecting stations i and j

d_{ij} variable unit flow time on arc (i,j), or variable passenger travel time cost on guideway connecting stations i and j

d_w passenger waiting time per time unit

d_t passenger transfer time

c_{ij} variable cost of operating and amortizing a vehicle on arc (i,j)

a_{mij} incidence designator for arc $(i,j) \in$ route m, $a_{mij} = 1$ if (i,j) m, $a_{mij} = 0$ otherwise

r^k_i net supply of commodity k at node i per operating time period, distributed uniformly during that period; $r^k_i > 0$ at the origin for commodity k, $r^k_i < 0$ at the destination node, and $r^k_i = 0$ elsewhere, the number of passengers of commodity k who wish to board at station i g capacity of a vehicle

U a large number.

The first summation in the objective function represents the arc fixed costs. Each p_{ij} would normally be the equivalent cost of owning and maintaining the exclusive one-way guideway connecting stations i and j during the operating time period. Because of the arc symmetry equations each p_{ij} represents one-half the cost of owning and maintaining the two-way guideway between i and j.

The second summation in the objective function represents the passenger travel costs. The cost d_{ij} reflects primarily the arc traversal time but also includes stopping times and passenger processing times. It is assumed that travel is on exclusive guideways or uncongested streets; hence, travel time is not a function of flow. All flow units of a commodity follow the same path.

The third part, the passenger waiting costs, reflects the assumption that passengers of a commodity k assigned on route m are distributed uniformly on evenly spaced vehicles on route m. Actually, this simplification results in a lower bound for the true waiting costs.

The transfer cost d_{ij} is assumed constant for each such transfer. If the actual timetable schedule to be eventually constructed is designed to provide for coordinated schedules on connecting routes, d_t will be relatively small. In such a case the waiting costs d_w incurred by passengers will also be small, and d_t can be adjusted downward so that $(d_t+(d_w/y_m))$ is a reasonable representation of the total time and nuisance involved.

In the last part of the objective function, vehicle operating costs, the cost c_{ij} is based on arc traversal time and power consumption. It is assumed that vehicle capital costs are recovered on a mileage basis and therefore included in the c_{ij}.

The multicommodity flow requirements equations are expressed for each commodity and each node. The vehicle capacity equations limit the total flow on each arc of each route to the total vehicle capacity provided during the operating time period. The arc feasibility constraints prevent vehicles from being assigned to closed arcs, and the symmetry arc property equations reflect the assumptions concerning two-way guideways.

To determine the waiting time occurrences for a particular commodity k, the number of times that the flow of commodity k into a node does not equal the flow of the commodity out of the node on the same route is counted. Since all flow units of a commodity follow the same paths, any imbalance must represent either vehicle boarding or deboarding by all passengers of commodity k. The absolute value operator in the delay time constraints count such imbalances for each commodity on each route, and since boarding and deboarding are paired, the result is divided by two.

The transfer time occurrences are counted in a similar manner except that the summation is taken over all routes for each commodity, and the first boarding-deboarding imbalance pair is not considered a transfer.

For a maximally connected network of ten stations and with each vehicle route limited to touching five stations, the above formulation results in approximately

5,200,000 continuous variables

1,100,000 integer variables

450,000 constraints.

Sharp's Algorithm

Even though the extensions made in this research interfere only with certain parts of Sharp's algorithm, an understanding of his entire program is needed to understand this work fully. Explanations will be given on all phases of the algorithm, and they will be coupled together at the end of this section to give the reader a working knowledge of Sharp's solution procedure.

The solution procedure consists of two algorithms, the second imbedded in the first. The first selects the set of guideways to be open, the second determines the best flow assignments, vehicle routes, and route service frequencies for this set of open arcs.

Guideway Insertion-Deletion

The guideway insertion-deletion algorithm is designed to minimize the guideway fixed costs and variable flow costs subject to flow requirements, guideway feasibility, integrality and nonnegativity.

The system starts with an initial set (A1) of open guideways and flow is assigned by the flow and route algorithms to minimize the variable costs with respect to the fixed set of guideways. To minimize the sum of fixed and variable costs, trade-offs are calculated to determine if fixed costs can be raised or lowered to decrease total costs. Hence, guideway improvement parameters are calculated to reflect this trade-off. These are called insertion parameters for closed guideways and deletion parameters for open ones. An insertion parameter is the algebraic sum of a positive change in fixed costs, p_{ij}, from adding guideway (i,j) and a negative change in variable costs, d_{ij}, from rerouting a flow. Deletion parameters include a negative change in fixed cost, p_{ij}, from removing guideway (i,j) and a positive change in variable costs, d_{ij}, from rerouting a flow.

A negative improvement parameter indicates that the projected guideway inserted or deleted will cause a cost decrease due to the change in fixed cost plus variable flow costs. Since computation of exact improvement parameters would involve consideration of the vehicle operation cost, c_{ij}, the passenger delay time costs, d_w and d_{ij} and constraints regarding vehicle capacities, approximate improvement parameters are used, and thus an indicated cost decrease must be verified.

Route Configurations Allowed

The number of possible routes connecting even a small network is large. For example, in a maximally connected system with ten stations, the number of routes of length ten arcs or less is about 25,000. Therefore, in order to reduce the computation time, certain specifications with regard to route length, stations included, and service frequency are imposed to restrict the number of combinations allowed.

There are two types of possible routes allowed in this program, loop routes and reversal routes. In a loop route (Figure 2) each station included has exactly one arc entering and one arc leaving, thereby forming a loop. Traffic can flow only in one direction in a loop route.

Reversal routes (Figure 3) are routes which reverse on themselves; that is, flow encounters the same nodes in reverse order from destination to origin as from origin to destination. Thus traffic can flow in both directions on a reversal route.

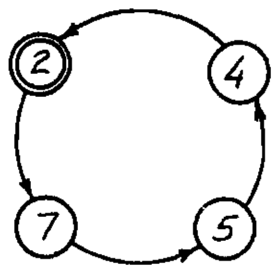


Figure 2. Loop Route

Figure 3. Reversal Route

Vehicle service frequency also follows certain specifications. For loop routes the vehicle service frequency must be constant over the route, i.e., there must be the same frequency of vehicle service on all arcs. For reversal routes two rules are in effect. The first is that there is the same frequency of vehicle service between station i and station j as there is between station j and station i (in the same route). The second rule is that the service frequency must either remain the same or decrease as the route progresses from the origin station to the destination station. Only integer values are considered for service frequency, in order to facilitate subsequent vehicle assignment which is not part of this research.

All reversal routes must start at a major node and all loop routes must contain at least one major node. A major node is a station at which vehicles may be stored during off-peak hours, and they are determined prior to execution of the program unless they are changed in reversing routes as described in Chapter IV.

Vehicle service frequency is expressed in number of vehicles per operating time period. Consider the following route in Figure 4.

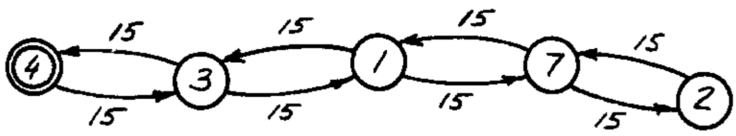


Figure 4. Reversal Route with Service Frequencies

The frequency between stations four and three is fifteen vehicles per time period. The next arc's service frequency must be equal to or less than fifteen, and so on, until the destination station where the symmetry rule comes into effect.

Figure 5 gives an example of a loop route and its service frequency; station four is considered the major node.

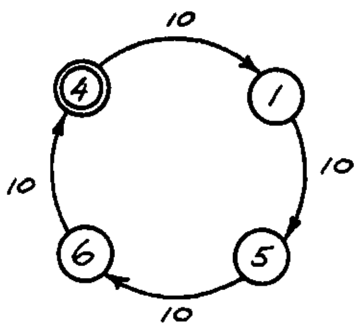


Figure 5. Loop Route with Service Frequency

Revised Networks

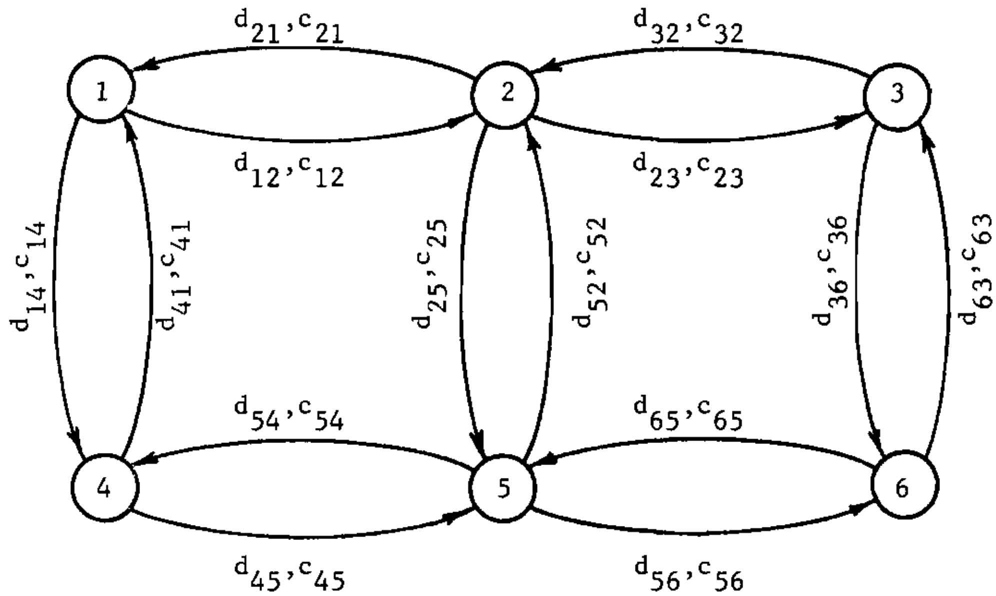
In constructing a network model of a transit system a means has to be incorporated to account for all elements of time spent by passengers. Passenger travel time itself is simply the distance required to travel divided by the vehicle operating velocity. However, travel time is only a portion of the total time, and the waiting and transfer times are usually the deciding criteria for customers using the system, since travel time for private vehicles and the transit system's vehicles should be closely related. How long the passenger has to wait initially to board his desired vehicle route and how many transfers are involved prior to his final deboarding are usually essential items. Waiting and transfer time are modeled as "travel" along dummy arcs. A revised network is constructed by assigning dummy nodes to account for origin stations, waiting stations, destination stations, and route stations, and dummy arcs to connect these stations. There is exactly one origin station and one destination station associated with each node and there is a route station corresponding to each node every time it is used in a route.

All passengers initially start at the origin station corresponding to their origin node. They then travel to the route station corresponding to the route desired for that node, the length being inversely proportional to the frequency of service on the route. Traveling from one node to the next in the route is just travel time until deboarding the route, either to transfer or terminate, with deboarding taking no time. If a passenger terminates here, his total time is the initial boarding or waiting time plus his travel time, and he proceeds to the destination node associated with this station. If he needs to transfer, he proceeds to the origin node associated with this station and incurs a transfer cost. He then boards his next route, incurring another waiting cost.

Consider the network in Figure 6 containing six stations, fourteen one-way guideways and three one-vehicle routes. To construct the revised network for this system, twenty-three nodes are needed: six origin nodes, six destination nodes, three route nodes associated with the three stations in route a, four route nodes associated with the four stations in route b, and four nodes associated with the four stations in route c.

Routes a and b have two-way arcs between their route nodes since they are reversal routes and route c has only one-way arcs connecting its route nodes since it is a loop route. There are two-way arcs connecting each origin node to each of the route nodes associated with that station and one-way arcs going into the destination node from the route nodes associated with that station. Origin nodes are labeled by placing the letter "o" after the station number; thus, the origin node for station one is 1o. Destination nodes are labeled by placing the letter "z" after the station number; thus, for station one the destination node is labeled 1z. Route nodes are labeled by placing the route identifier after the station number, thus station one in route b is labeled 1b. The revised network for the system in Figure 6 is given in Figure 7.

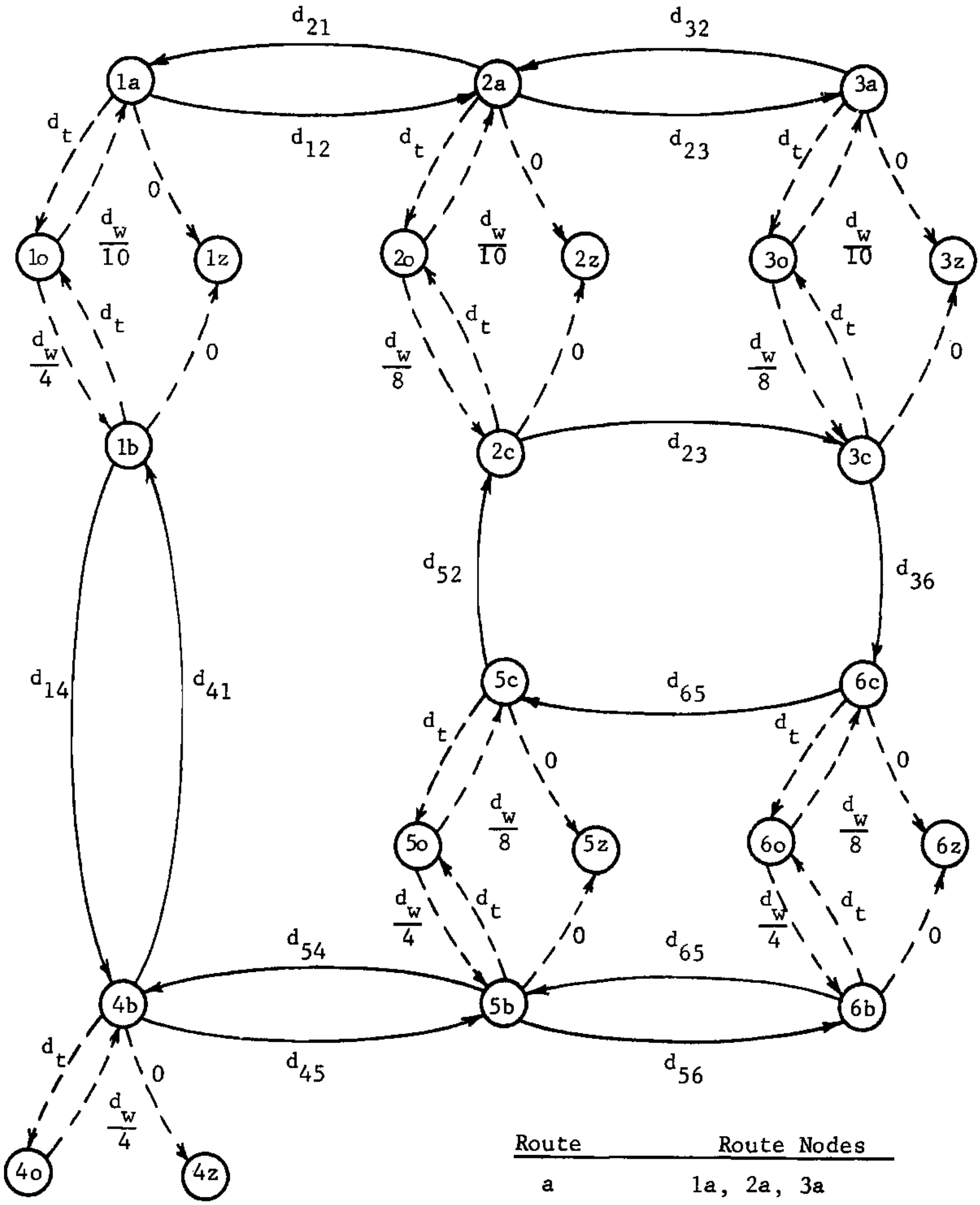
Consider a passenger traveling from station four to station three. With the present routes he has two logical choices, either to use route b to travel from station four to station one, then to transfer to route a and to travel from station one to station two to station three, or to use route b to travel from station four to station five, to transfer to route



d_{21},c_{21} Distance from station two to station one, relative cost of fixed guideway from station two to station one.

Route	Stations	Service
a	1-2-3-2-1	$y_a = 10$
b	1-4-5-6-5-4-1	$y_b = 4$
c	3-6-5-2-3	$y_c = 8$

Figure 6. Original Network



Route	Route Nodes
a	1a, 2a, 3a
b	1b, 4b, 5b, 6b
c	2c, 3c, 6c, 5c

Figure 7. Revised Network

c and to travel from station five to station two to station three. The first path incurs a waiting time, $d_w/4$ to board route b, a travel time, d_{41} to travel from station four to station one, a transfer time, d_t, to transfer through node ten, a waiting time, $d_w/10$ to board route a at 1a, and travel times, d_{12} and d_{23} to travel to node 3a, and zero deboarding time. The total time incurred is $d_w/4 + d_{41} + d_w/10 + d_{12} + d_{23}$. The costs for the second path are $d_w/4$ to board route b at node 4b, travel time d_{45} to travel to node 5b, d_t to transfer to node 5o, $d_w/8$ to board route c at node 5c, travel time of d_{52} and d_{23} to travel from station five to station two to station three, and zero deboarding time for a total of $d_w/4 + d_{45} + d_t + d_w/8 + d_{52} + d_{23}$. The passenger is initially assigned to the path that gives him the minimum total cost.

In the revised network in Figure 7 all the waiting times, travel times, and transfer costs are represented explicitly as arc flow costs. These costs are "variable" costs in the cost accounting sense. The selection of a least-cost path for a single commodity then becomes a shortest path network problem, and the assignment of all passenger commodities becomes a capacitated, minimal-cost multicommodity network flow problem.

Route Insertion-Deletion

The route insertion-deletion algorithm was developed to account for the vehicle amortization and operating costs c_{ij} in the objective function. Fixed arc costs only apply to arcs utilized by vehicles. Boarding, transfer, and terminal arcs have zero fixed costs.

The problem is to select a set of fixed-cost arcs for inclusion in a network to minimize the total of fixed plus variable costs. An arc insertion-deletion algorithm is used to obtain a solution, with arc improvement parameters leading to route improvement parameters, and arc insertion and deletion becoming route insertion and deletion and route modification. The details of the algorithm and computation of improvement parameters are presented later in this chapter.

Penalty-Cost Multicommodity Flow Assignment Routine

The penalty-cost multicommodity flow assignment routine is designed to assign all commodities to their shortest paths such that the total flow over each arc in the system is within capacity. For commodities assigned to infeasible arcs, the next shortest path is chosen and this is where arc infeasibility parameters are generated.

Initially, all infeasibility parameters are set to zero, and each commodity is assigned to its shortest path disregarding arc capacities. If there are any infeasible arcs, we determine the set of commodities, K1, which flow over an infeasible arc. For these commodities we compute the shortest and second shortest path lengths, p_1 and p_2, respectively, the number of infeasible arcs n_1 in the shortest path, and the number of inadequate arcs n_2 in the second shortest path, such that $n_2 < n_1$. An inadequate arc is one that would be infeasible if the commodity were assigned to it. If n_2 is not less than n_1 for the second-shortest path, we compute the third-shortest path, etc. If there is no n^{th}-shortest path such that $n_2 < n_1$ for all commodities, $k \epsilon K1$, the system is infeasible. For each commodity $k \epsilon K1$ we compute

$$u_k = \frac{p_2 - p_1}{n_1 - n_2} ,$$

and select $k^* \epsilon K1$ such that u_{k^*} is minimized. Let eps $= u_{k^*}$, and commodity

k^* becomes the next commodity to be reassigned. Increase each infeasible arc by eps and add -(excess flow $\times$ eps) to the previous infeasibility parameter for each infeasible arc. We reassign commodity k^* to its second-shortest path, and we make the corresponding changes in total flow to each arc included in either of these paths.

We determine the set of infeasible arcs and continue this process until all arcs are feasible or the system is declared infeasible. This routine obtains a good, but not necessarily optimal, solution to the minimal-cost multicommodity flow problem. An important by-product is the computation of the arc infeasibility parameters, representing the cumulative infeasibility of an arc during the process of changing from an initial infeasible flow assignment to a final feasible assignment, measured in terms of variable arc flow costs.

Arc Insertion Improvement Parameters

The arc infeasibility parameter provides a rough estimate of potential savings if the capacity of the arc is increased. For example, assume the shortest path for commodity k, with travel demand 100, contains one infeasible arc ($n_1=1$) carrying an excess flow of 100 units, and that the second-shortest path is $p_2-p_1 = 20$ units longer. Increasing the one infeasible arc in the first path by twenty units will cause the commodity to switch to the feasible second-shortest path at a total variable cost increase of $(20) \times (100) = 2000$ units. Likewise, if the first path contained $n_1 > 1$ infeasible arcs, then each of these arcs needs to be increased by $(p_2-p_1)/n_1$ to cause the commodity to switch paths for the same variable cost increase. Incrementing the arc infeasibility parameters serves to record all such potential savings.

Increasing the capacity of an arc causes fixed costs to increase; however, it is likely that variable costs will decrease due to shorter paths for some commodities. These arc infeasibility parameters are estimates of the overall cost changes when specific arcs are inserted. In the route insertion-deletion algorithm, the required arc insertion improvement parameters are defined to be these arc infeasibility parameters.

Route insertion improvement parameters are obtained by summing the insertion parameters for all non-dummy arcs contained in each route (no parameters are accumulated for boarding, transfer, or terminal arcs). (These parameters are also summed over all routes on each guideway to calculate insertion parameters in the guideway insertion-deletion algorithm described earlier.) How much to increase capacity is determined by the excess flow on the arc, which is added to the arc infeasibility parameter and also cumulated on guidways.

Increasing Service on Existing Routes

The first method of "inserting" a route is to increase service on an existing route. The route selected for increased service is the one with the best insertion improvement parameter, loop route parameters obtained by summing the arc parameters for all arcs in the route.

For reversal routes, either the whole route or segments can have service increased. Consider the following reversal route in Figure 8.

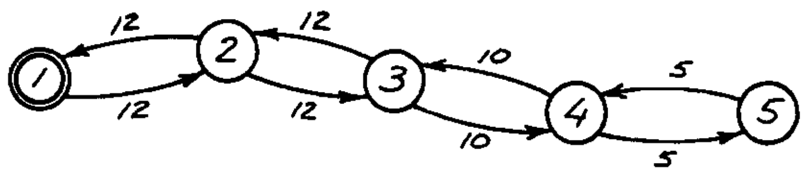


Figure 8. Reversal Route

Service increases can be made on the following segments:

1-2-3-4-5, 1-2-3-4, 1-2-3, 1-2,

3-4-5, 3-4,

4-5.

Increases starting at station one can be any size. Increases cannot start at station two since capacities on the first two-way arc would still limit the capacity of the entire route to twelve vehicles. Increases starting on the third arc are limited to two vehicles and increases starting at the last arc are limited to five vehicles.

The parameters of all sections of a reversal route are compared and the most negative is selected. If two sections of unequal lengths have the same improvement parameters, the shorter section is chosen since fixed costs will increase less for the same indicated variable cost decreases. Service is then increased by one vehicle per time period and the multicommodity flow assignment algorithm is called to assign flows and determine total fixed and variable costs.

If the result is a cost decrease, the changes are kept and new parameters are computed to determine if more service is needed. If the previous set was infeasible and costs are increased then it is assumed that the infeasibility is reduced due to increased capacity, and the new service levels are kept. If the previous assignment was feasible and total costs are increased, then the previous routes, service frequencies, and arc improvement parameters are reestablished; guideway appending is entered upon encountering this condition.

Appending Guideways to Existing Routes

This section of the Sharp's algorithm attempts to extend reversal routes by appending arc pairs to the ends of the routes. (The end of a reversal route is the node farthest from the route's major station.) The previously computed guideway insertion improvement parameters (sums of arc insertion parameters) are used to determine which guideways should be appended, the guideway with the most negative insertion parameter being selected. A preliminary test is made against the fixed cost (the vehicle operating cost) for the guideway pair that constitutes the route extension. The routes are then checked to determine if that guideway can be appended. If it cannot be appended to any reversal route, it is placed in a hold list and the guideway with the next most negative parameter is selected. When a guideway is appended any corresponding zero-vehicle route is deleted. If there exists more than one route to which the guideway can be appended, then the route is selected which most closely corresponds to the desired capacity of the new section. The capacity assigned to this section cannot exceed the service frequency of the previous section; thus, zero-vehicle routes are not considered as candidates for extension by appending. The multicommodity assignment algorithm is then entered to assign flow and determine total costs.

An overall cost improvement leads to computation of new parameters, a clearing of the hold list, and more attempted appending. A cost deterioration causes the previous routes, flow, and parameters to be reestablished and the guideway pair placed in the hold list. This phase is exited either when the preliminary test fails or when the hold list exceeds its maximum length.

Route Construction

The third method of inserting a route is to construct a new route. There are three restrictions which have to be observed, major station starting point, reversal or loop route form, and route length.

The program selects a major station at the end of an arc with an extreme parameter value. An out-tree is then constructed, with the major station as origin, such that no branch contains more than one half the maximum number of route arcs. From here, the two types of routes, loop or reversal, can be constructed. The reversal route consists of doubling back along the corresponding guideway arcs. For loop routes, another tree, an in-tree, is constructed from each node of the out-tree back to the origin node, with no node being contained more than once in an in-tree and once in the out-tree.

The route selected for insertion is the one with the most negative length, or the best insertion improvement parameter. This route is first compared with existing routes to check for duplication, in which case service is increased on that existing route. A check is also made to see if any zero-vehicle routes can be deleted. The multicommodity flow assignment algorithm is then entered and costs are checked. A success leads back to the phase on increasing service on existing routes while a failure leads to route deletion.

Route Deletion

To decrease the fixed costs associated with a route without significantly increasing variable costs, excess service can be decreased. Passenger travel and transfer times remain the same, passenger waiting times increase, and vehicle operating costs decrease.

There are two phases to this process. First, the excess capacity of the route and route sections are calculated by determining the minimum excess capacity of arcs in the route or section and choosing the maximum of this over all routes and sections. The route or section selected has its service reduced by this amount and total fixed and variable costs are computed for this revised network. If an improvement occurs the process is repeated; otherwise the second phase is entered.

The second phase sums the excess capacity of each arc in every route and picks the route with the maximum total excess capacity. Each arc of this route has its service decreased by one unit per time period. If overall costs decrease the process is repeated, otherwise the algorithm enters the "arc deletion improvement parameter" section.

Arc Deletion Improvement Parameters

Arc and route insertion improvement parameters were a natural by-product of the multicommodity flow algorithm; however, deletion parameters are not so easily obtained since there is no obvious way to estimate increases in variable costs due to reducing arc capacities.

Lacking parameters, we institute a direct search, reducing service separately on each arc of a route by one vehicle per time period, constructing the revised network and computing the usual arc infeasibility parameters. Arc deletion parameters are then defined to be the savings in vehicle operating cost minus the arc infeasibility parameter. Since the arc infeasibility parameter of an arc is a measure of potential savings in variable costs obtained by increasing the arc's capacity, and service has been reduced, it represents potential variable cost increases due to reducing capacity. It is subtracted to change the sign, which was negative, to indicate potential cost increases, and the savings in vehicle operating costs, representing fixed cost savings, are added; the result, finally, is an arc deletion improvement parameter.

Arc deletion parameters are then used to determine which section of a reversal route to reduce by maximizing the sum of the arc deletion parameters over the section.

Summary of Sharp's Solution Procedure

The research problem involved is to determine which fixed-cost guideways should be constructed and which vehicle routes and service frequencies are needed to satisfy total demand at a minimum total cost. The procedures described above may be summarized as four main phases:

(1) guideway insertion-deletion algorithm

(2) route insertion-deletion algorithm

(3) penalty-cost multicommodity flow routine

(4) shortest path assignment.

The shortest path assignment routine assigns all commodities to their shortest paths, based on the revised network, disregarding arc capacities. The penalty cost multicommodity flow routine switches commodities from infeasible shortest paths to feasible second- or third-shortest paths.

Once all commodities flow on feasible paths, the system is declared feasible and arc improvement parameters are calculated. The route insertion-deletion algorithm is then entered to check for insufficient or excess capacity or to determine if arcs can be appended or new routes constructed. When no more cost reduction improvements can be made, the guideway insertion-deletion algorithm is entered.

The guideway insertion-deletion algorithm changes the set of open guideways either by insertion or deletion of a guideway, if it is found total costs can be improved from the subsequent new route set and passenger flows. Approximate guideway insertion improvement parameters are used here, based again on the idea of second-shortest paths.

Figure 9 shows the generalized flowchart of Sharp's solution procedure. A successful change in any phase results in a repeat of that phase, indicated by a dotted line. A solid line indicates normal progress of the program or the exit from a phase after an unsuccessful change.

Shortcomings of Sharp's Algorithm

Sharp's algorithm gives good solutions to small problems, but the final solutions of larger problems are heavily dependent on the initial starting points. It was this dependency that gave rise to the present research.

One problem with twelve stations had a final solution consisting of twelve routes, including nine two-arc routes and three four-arc routes. This solution forced passengers traveling to a station more than two arcs away to transfer at least once. If these routes could have been combined at common nodes then passenger transfers would have been greatly reduced. It was apparent that Sharp's algorithm needed more powerful route restructuring capability.

Another difficulty in Sharp's solution procedure is duplication of routes. A test problem's solution resulted in six routes; however, there were two cases, listed below, of essentially-identical route pairs:

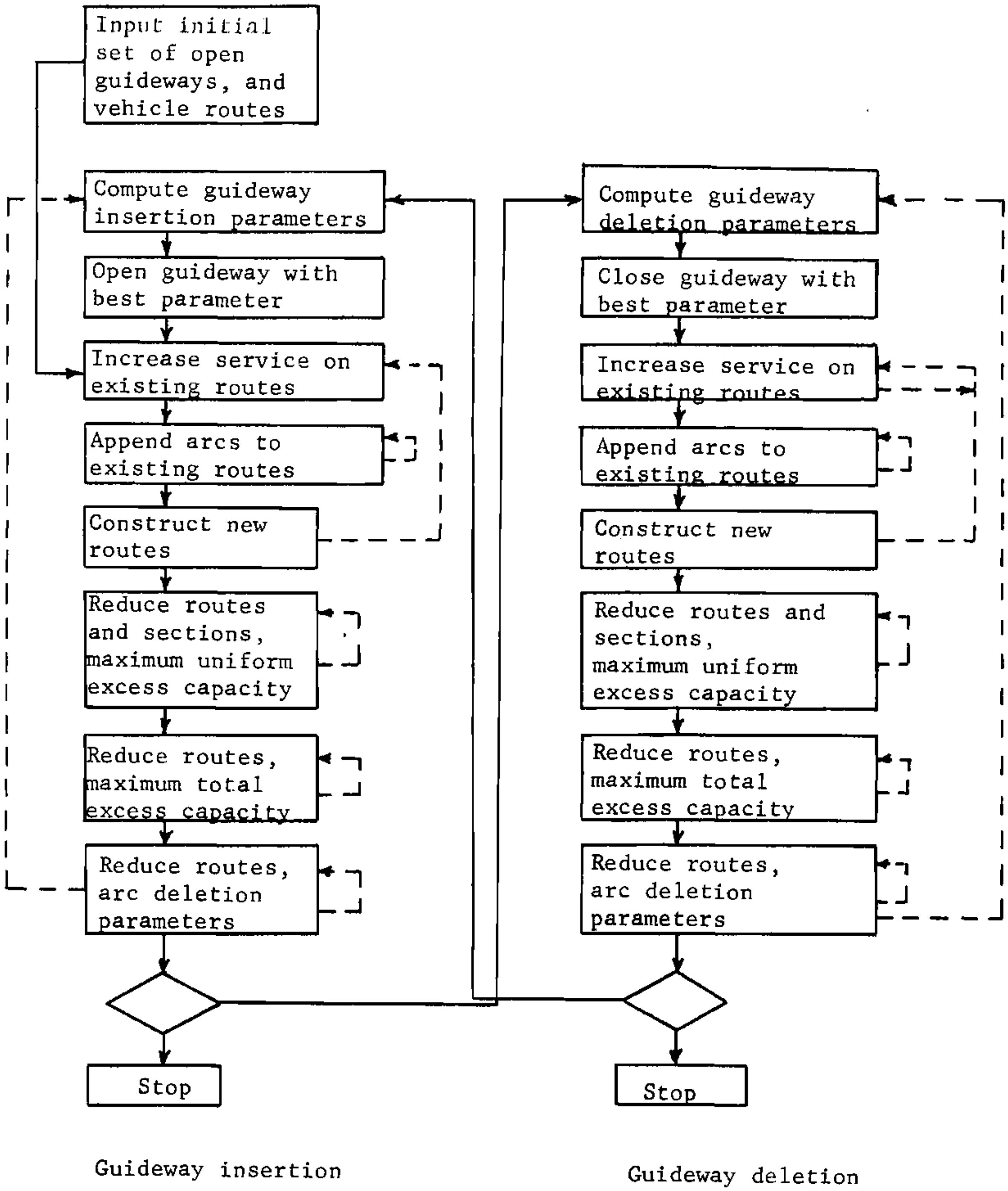


Figure 9. Flowchart of Solution Procedure

Route	Stations
1	6-4-3-1-3-4-6
3	1-3-4-6-4-3-1
2	1-3-4-5-4-3-1
5	5-4-3-1-3-4-5

Route one is identical to route three but in the reverse direction, as are routes two and five. A method was apparently needed to check for this duplication which results in excessive passenger waiting time. Sometimes Sharp's algorithm gives a route identical with a portion of another, which should also be guarded against.

Sharp's procedure attacks guideway insertions and deletions on a one-at-a-time basis. This is sometimes ineffective since inserting a guideway might result in excess capacity to an area, whereas if one could be deleted at the same time the balance would be preserved. Similarly, a guideway cannot be deleted if it would isolate a station or set of stations from the remainder. Thus, if a guideway were added at the same time, the deletion could take place. Cases also arise when inserting or deleting a guideway singly causes increases in total cost, but the combined effect of insertion and deletion results in lowered costs. This gave rise to the multiple guideway insertion-deletion portion of this research.

CHAPTER IV

SOLUTION PROCEDURE

The improved solution procedure reported here is basically the same as Sharp's program with changes made in the guideway insertion-deletion algorithm and in the route structuring algorithm. The improved procedure consists of three main algorithms, the second imbedded in the first and the third imbedded in the second. The first selects from the set A of possible arcs that subset A1 of arcs which are to be currently open. The second determines the best flow assignments, vehicle routes, and route service frequencies for the given subset A1 of arcs, using the subroutines from Sharp's program. The third algorithm, which comprises the main work of this research, is a route restructuring algorithm. It determines whether routes should be combined or reversed to save passenger travel, transfer and waiting time costs, and vehicle operating costs.

The first two algorithms are of the arc insertion-deletion type used successfully by Scott (12) and Billheimer (4). They rely heavily upon finding the shortest and second-shortest paths for commodities and the subsequent computation of improvement parameters. Both also require a capacitated, minimum cost multicommodity flow algorithm as a subroutine.

Guideway Insertion-Deletion

In Sharp's guideway insertion-deletion algorithm, if a guideway insertion does not result in a cost decrease, then the insertion is rejected and guideway deletion is attempted next. However, a step-by-step reduction in cost does not always guarantee a minimum, especially with the cost structure present here. That is, a cost increase at one iteration may eventually allow a further decrease at the next iteration or further downstream. Sharp's guideway insertion-deletion algorithm has been slightly modified accordingly.

This modified algorithm first computes guideway insertion parameters and inserts the most negative one if one exists. After insertion of the guideway it checks the cost, but regardless of whether a decrease occurred or not, it switches to guideway deletion with the inserted guideway still in the set A1. If the final result of both a guideway insertion and guideway deletion is a net cost decrease, the changes are kept. If the final result is a net increase but guideway insertion resulted in a cost decrease, then the guideway inserted is left in but the one deleted is restored to the set A1.

If either the final result of the guideway insertion-deletion or the guideway insertion alone was a cost decrease, the algorithm attempts another insertion-deletion. However, if both resulted in a cost increase, then deletion only is attempted. If this is successful then the algorithm starts again at the guideway insertion-deletion. If the deletion-only phase resulted in a cost increase, then the algorithm stops.

Each time a guideway is added to or removed from set A1, the second algorithm (identical to the corresponding portions of Sharp's algorithm) is executed to obtain the best flow assignment, vehicle routes, and service frequencies.

Route Restructuring

The third algorithm, route restructuring, solves the same problem as Sharp's route scheduling algorithm, but uses as its starting point the solutions generated by his algorithm and attempts to improve on them by various route manipulations. There are many ways of manipulating or restructuring routes, some of these being: combining routes, splitting routes, reversing routes, or a combination of the three. Combining routes can be done in a number of ways. A route can be attached to the beginning, end, or middle of another route, or the route can be combined with an identical segment of another route to eliminate duplication.

Consider the following set of routes (Figure 10):

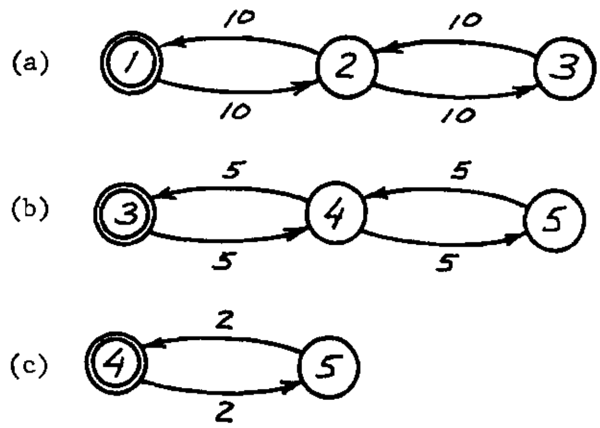


Figure 10. Route Set 1a

Routes a and b can be combined directly to form the following set (Figure 11):

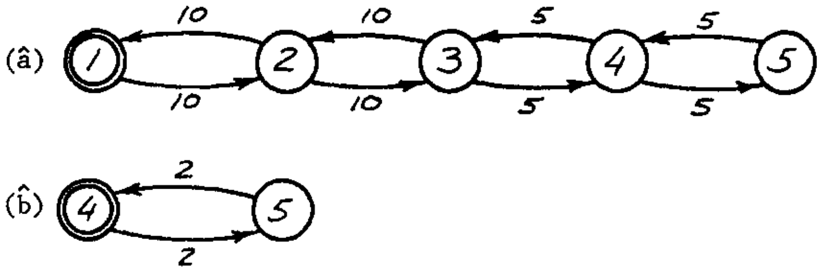


Figure 11. Route Set 1b

Notice that the service frequency of route a was greater than the service frequency of route b therefore causing route b to be added to the end of route a. In order to have placed route b first, the frequency of all its arcs must be increased to at least ten vehicles per time period to match the first arc of route a. This combination saves transfer and delay time cost to any passengers who use both routes a and b, for example, passengers who travel from station one to station four. Since there is no vehicle service increase there are no cost increases associated with this combination.

A further combination (Figure 12) can be made by adding route b̂ to route â.

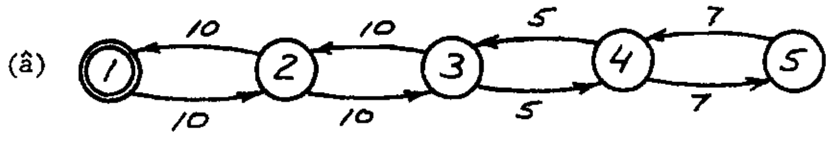


Figure 12. Route Set 1c

Since a direct addition of service frequencies of route b̂ on the identical portion of route â gives an invalid route, this newly formed route must be modified to conform to the service frequency specifications given. This is achieved by increasing the service between stations three and four (Figure 13) by two units.

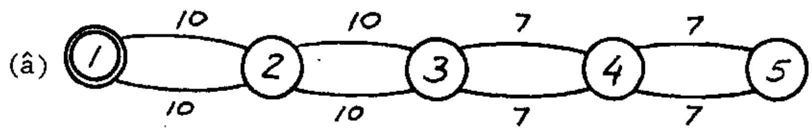


Figure 13. Route Set 1d

Savings resulting from this combination are due to passengers who were transferring to or from route $\underline{\hat{a}}$ to route $\underline{\hat{b}}$ and by passengers waiting at stations four or five for either route $\underline{\hat{a}}$ or $\underline{\hat{b}}$. Cost increases occur due to the unnecessary increase in service frequency of two units between stations three and four. If the decrease in passenger transfer and delay time costs is greater than the increase in vehicle operating cost, the change is incorporated; otherwise, the system reverts back to the previous set of routes. No cost savings are attributed to a reduction in the number of routes.

Another combination possible is reversing a route prior to adding it to the end of a route or combining identical arcs and stations (Figure 14). No routes in this set can be directly combined since the destination stations of all routes do not match any origin stations of another route. There are also no identical portions of routes in a forward direction. However, if route $\underline{b}$ can be reversed it can be added to route $\underline{a}$ with the vehicle service frequency of route $\underline{b}$ adjusted prior to the combination (Figure 15).

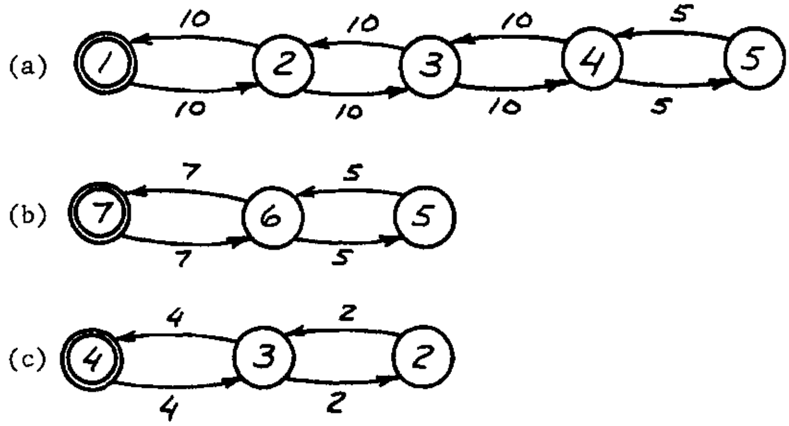


Figure 14. Route Set 2a

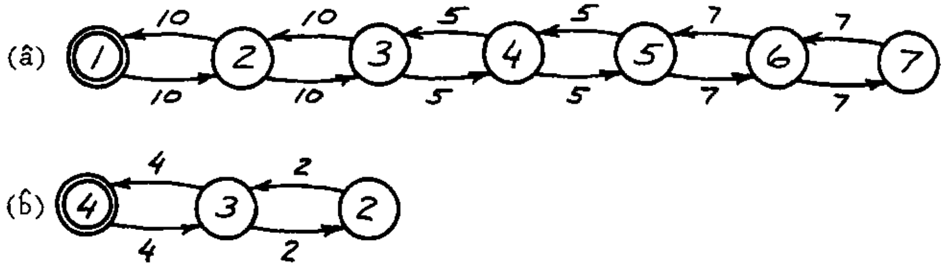


Figure 15. Route Set 2b

Since this combination gives an invalid route the frequencies must be adjusted on the arcs between stations four and five (Figure 16).

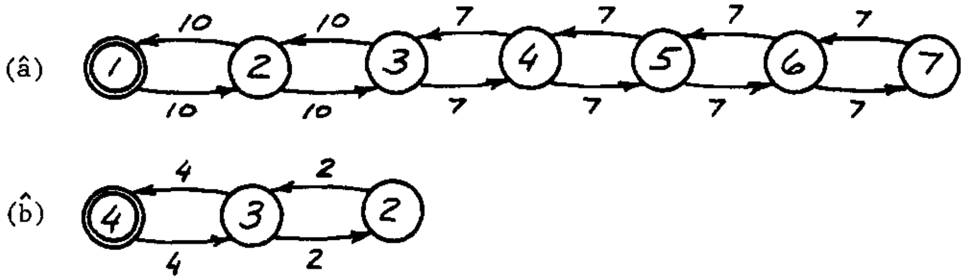


Figure 16. Route Set 2c

Cost increases from this combination are from increased vehicle service between stations three, four, five, and six of two vehicles in each direction. Cost decreases occur by lower waiting times at stations three, four, five, and six and by deleting the transfer time between routes $\underline{\hat{a}}$ and $\underline{\hat{b}}$.

Route $\underline{\hat{b}}$ is identical to a portion of route $\underline{\hat{a}}$ but in a reverse order. When reversing a route to add <u>into</u> another route, the vehicle service frequency is not adjusted prior to the combination. This is because the imbalance of vehicle service might be absorbed by an imbalance in the other route. That is, the extra two vehicles between stations three and four in route $\underline{\hat{b}}$ are offset by the three vehicle decrease in route $\underline{\hat{a}}$ from arc (2,3) to arc (3,4) (Figure 17).

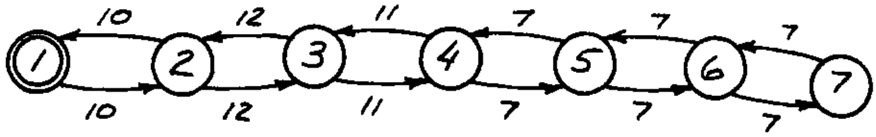
($\hat{a}$)


Figure 17. Route Set 2d

The adjustment must now be made to arc (1,2) by increasing its vehicle service by two units (Figure 18).

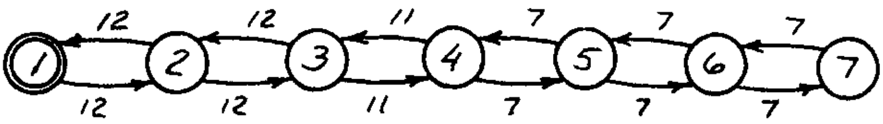
($\hat{a}$)


Figure 18. Route Set 2e

Again cost increases occur from increased vehicle service frequency while cost decreases occur from decreased passenger transfer and delay time costs.

Another possible restructuring scheme is to reverse a route if there is a savings involved. Due to the service frequency structure, many times some arcs are "padded" with vehicles to satisfy a heavy demand further downstream. With the following flow requirements between these stations (Figure 19):

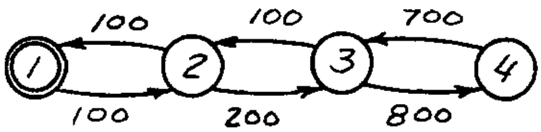


Figure 19. Route Set 3a

the service must be assuming vehicle capacity of one hundred (Figure 20):

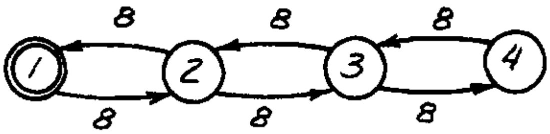


Figure 20. Route Set 3b

whereas there actually needs to be only one vehicle between stations one and two, two vehicles between stations two and three, and eight vehicles between stations three and four. Thus if the route were reversed, that is, if the major station were station four instead of station one, a total of twenty-six vehicles per time period would be saved. This savings in vehicle operation cost must be weighed against increases in passenger waiting time cost (Figure 21).

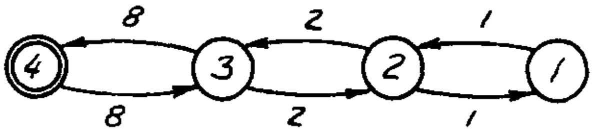


Figure 21. Route Set 3c

If this results in a cost savings, the major node criterion is waived for this route.

Routes cannot be combined if there is more than one common node between them. One common station is needed as a connecting station, but more than one indicates that the route crosses itself, which is not allowed. Figure 22 illustrates this set of routes.

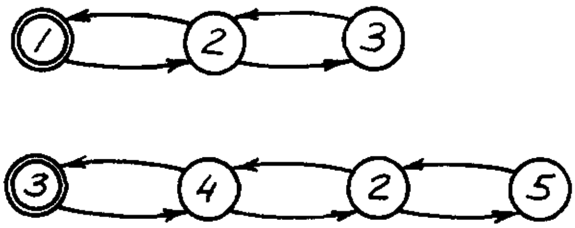


Figure 22. Route Set 4a

Combining the two routes gives the crossing at station two, thereby forming a cycle in the middle of the route (Figure 23).

or

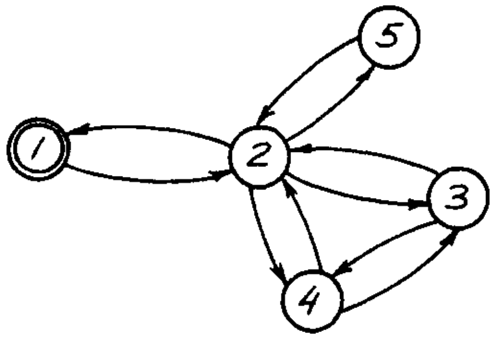


Figure 23. Route Set 4b

To incorporate these restructuring features into the solution procedure, a series of FORTRAN V subroutines were written. Eight of the subroutines handle specific functions and are coupled together by the ninth subroutine. The functions performed are:

1) Check for a major station at the end of a route;

2) Put the beginning and ending stations in arrays;

3) Reverse a route to add to another route;

4) Combine two routes, one added to the end of another;

5) Combine common sections of two routes;

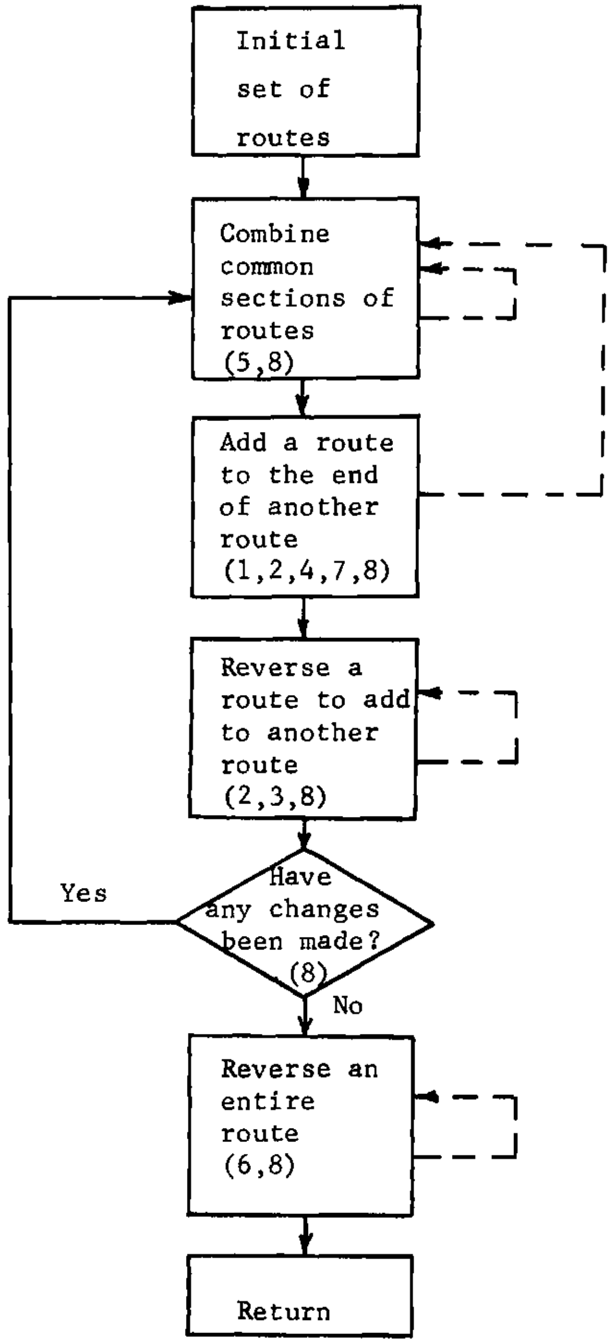
6) Reverse an entire route;

7) Print out the routes and service frequencies;

8) Check the cost of a route system;

9) Call the subroutines in the required order.

The eight subroutines can be called in a variety of combinations and some also are used singly in Sharp's program, such as reversing an entire route or combining identical route sections. A flowchart of the procedure used in the route restructuring algorithm is given in Figure 24. A dotted line represents a reduction in total costs followed by a repeat of the operation and the solid line indicates either an increase in total costs or normal flow of the algorithm.



(Numbers refer to functions performed.)

Figure 24. Flowchart of Route Restructuring Algorithm

CHAPTER V

COMPUTATIONAL RESULTS

Sharp's FORTRAN V program has been modified to incorporate the changes made in the guideway algorithm and to add the route restructuring algorithm into the route scheduling algorithm. This program uses approximately 36,000 words in core, 17,000 for instructions and 19,000 for data. The majority of the data are kept in integer form.

This chapter will show examples of the algorithms added to or modified in Sharp's program and will give results of the test problems run and comparisons with Sharp's results. Test problems are designated by symbols of the type C, C1, and C1a, where

- C designates the given station locations and travel demands, and the set of all possible guideways,
- 1 represents a set of open guideways for problem C,
- a represents a set of vehicle routes for problem C1.

Thus the guideway algorithm works with problem C and passes on to the route algorithm problems C1, C2, C3, etc. The route algorithm works on a problem of type C1 and passes on to the multicommodity assignment routine problems C1a, C1b, C1c, etc. The route restructuring algorithm works on the same type problems as the route algorithm and also passes problems of the type C1a, C1b, C1c, etc. on to the multicommodity assignment routine.

Passenger travel and waiting costs are both assumed to be $0.03 per minute, while a transfer cost equivalent to fifteen minutes is used. A vehicle operating cost of $1.00 per minute is used, vehicle capacity is one hundred passengers, and the operating time period is taken to be one hour.

Guideway Insertion-Deletion Algorithm

Test Problem E

Test problem E will be used to demonstrate the guideway insertion-deletion algorithm. The initial open guideways and beginning routes are shown in Figure 25 while the travel demands are given in Table 1.

The route algorithms take this initial data and solve for the best set of routes and passenger flows. At the completions of the route algorithm the improvement parameters for the closed guideways are computed, with only the negative ones being printed:

IMPROVEMENT PARAMETERS FOR CLOSED GUIDEWAYS

Arc	Parameter
(2,3)	-2371
(3,4)	-10833
(4,5)	-3214

Guideway Selected for Inclusion Is (3,4)

As can be seen, guideway (3,4) has the most negative improvement parameter and is thus selected as the guidway to enter the open set A1. The costs obtained from the initial set of open guideways and the route algorithms are:

Travel Time Costs	= $17721
Vehicle Operating Costs	= $ 5134
Guideway Fixed Costs	= $ 4550
Sum of Fixed Plus Variable Costs	= $27405

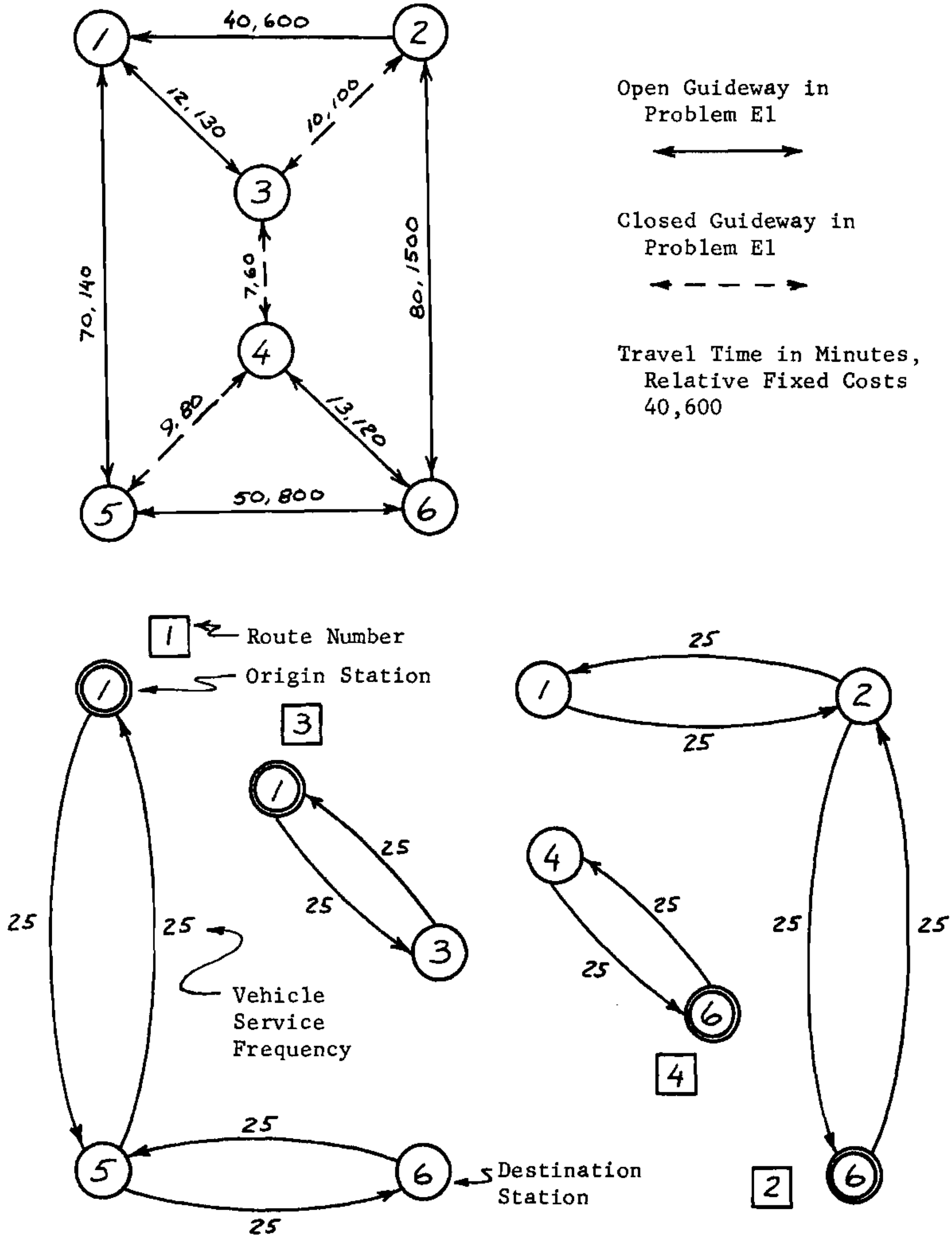


Figure 25. Problem E1 -- Open Guideways and Starting Vehicle Routes

Table 1. Problem E, Travel Demands

Commodity Number	Origin Station	Destination Station	Travel Demand
1	1	2	100
2	1	3	400
3	1	4	300
4	1	5	100
5	1	6	100
6	2	1	100
7	2	3	400
8	2	4	200
9	2	5	100
10	2	6	100
11	3	1	200
12	3	2	200
13	3	4	400
14	3	5	100
15	3	6	100
16	4	1	100
17	4	2	100
18	4	3	400
19	4	5	200
20	4	6	200
21	5	1	100
22	5	2	100
23	5	3	300
24	5	4	400
25	5	6	100
26	6	1	100
27	6	2	100
28	6	3	300
29	6	4	400
30	6	5	100

The current set of routes along with the new set of open arcs are passed to the route algorithms. The resulting costs from Guideway Algorithm Iteration 1 are:

Travel Time Costs	= $ 9942
Vehicle Operating Costs	= $ 2988
Guideway Fixed Costs	= $ 4610
Sum of Fixed Plus Variable Costs	= $17540

This insertion resulted in a net cost decrease of $9865 and will set a counter indicating that the insertion was good and should stay in, in case the combined effect of the guideway insertion-deletion results in a net cost decrease.

After a guideway insertion is completed the algorithm goes to guideway deletion. It is not allowed to look at the guideway just inserted which would recreate the previous set of guideways. The guideway selected for removal is (2,6) and this new set of guideways and the current set of routes are passed on to the route algorithms. The results of Guideway Algorithm Iteration 2, actually the first complete iteration of the guideway insertion-deletion algorithm, are:

Travel Time Costs	= $ 9590
Vehicle Operating Costs	= $ 2872
Guideway Fixed Costs	= $ 3110
Sum of Fixed Plus Variable Costs	= $15572

for an overall improvement of $11833.

Since the first complete iteration was successful, the algorithm attempts another guideway insertion-deletion starting with guideway insertion. The improvement parameters for the closed guideways are:

Arc	Parameter
(2,3)	-2090
(2,6)	-188
(4,5)	-3406

giving the guideway selected for inclusion as (4,5). This set of open guideways and the current set of routes are passed to the route algorithm which obtains a final cost of:

Travel Time Costs	= \$ 9579
Vehicle Operating Costs	= \$ 2886
Guideway Fixed Costs	= \$ 3190
Sum of Fixed Plus Variable Costs	= \$15655

for a cost increase of \$83. This guideway is allowed to remain in the set of open guideways until guideway deletion is completed; however, a counter is set indicating that this guideway should not be included if the results of the guideway insertion-deletion algorithm are unsuccessful. The guideway selected for removal is (1,5).

This new set of open guideways and the current set of routes are passed to the route algorithm, resulting in the following set of costs:

Travel Time Costs	= \$ 7078
Vehicle Operating Costs	= \$ 1892
Guideway Fixed Costs	= \$ 1790
Sum of Fixed Plus Variable Costs	= \$10760

for an overall improvement of \$4812 over the first guideway insertion-deletion algorithm. Thus arc (4,5), which created a cost increase when inserted by itself, is allowed to remain in the set of open arcs. In Sharp's algorithm, this arc would not be allowed to remain in the set but would be later reintroduced, remaining in. Thus a reduction of at least one guideway algorithm iteration has been achieved thus far.

The second guideway insertion-deletion algorithm was successful. Thus the program attempts another, again starting with arc insertion. The improvement parameter for the closed guideway is:

Arc	Parameter
(2,3)	-2521

thus arc (2,3) is chosen, being the only arc available with a negative improvement parameter. The new set of open guideways and current routes are passed to the route algorithms and result in the following set of costs:

Travel Time Costs	= \$4390
Vehicle Operating Costs	= \$1236
Guideway Fixed Costs	= \$1890
Sum of Fixed Plus Variable Costs	= \$7516

for an improvement of \$3244. Hence a counter is set specifying that this arc should remain open regardless of the outcome of the guideway insertion-deletion algorithm. Guideway deletion is attempted next and arc (5,6) is selected for removal. This new set of open guideways and the current set of routes are passed to the route algorithm, resulting in the following costs:

Travel Time Costs	= \$4566
Vehicle Operating Costs	= \$1182
Guideway Fixed Costs	= \$1090
Sum of Fixed Plus Variable Costs	= \$6836

for an overall improvement of \$3824. Thus the third guideway insertion-deletion algorithm was successful and a fourth is attempted, starting with arc insertion. However, there are no negative improvement parameters for closed guideways, indicating that no guideway should be attempted to be inserted. Hence the algorithm switches to arc deletion. The guideway selected for removal is (1,2). The new set of routes is passed to the route algorithm and the resulting costs are:

Travel Time Costs	= \$4377
Vehicle Operating Costs	= \$1126
Guideway Fixed Costs	= \$ 490
Sum of Fixed Plus Variable Costs	= \$5993

for an overall improvement of \$843. The program terminates here since there are no negative improvement parameters. Thus no guideways are candidates to enter, and no guideway can be deleted since doing so would cause at least one station to be isolated from the others, leaving an infeasible solution. Therefore, the program is terminated with results given in Figure 26.

This program obtained its solution after seven guideway iterations compared to Sharp's ten guideway iterations. The fixed guideways were the same in both solutions, with the majority of the savings being vehicle operating costs, a 19% decrease. Sharp's solution resulted in six routes; however, two are basically identical with two others. The basic structure of Sharp's routes, shown in Figure 27, is the same as the routes generated by this program, with this program reducing some of the route inefficiencies due to the route manipulations. A comparison of the guideway algorithm iterations is shown in Table 2.

Test Problem C

The multiple guideway algorithm was also used on two subproblems of test problem C, Cla and Clb. Both runs resulted in the same costs as the runs without the multiple guideway algorithm; however, the execution

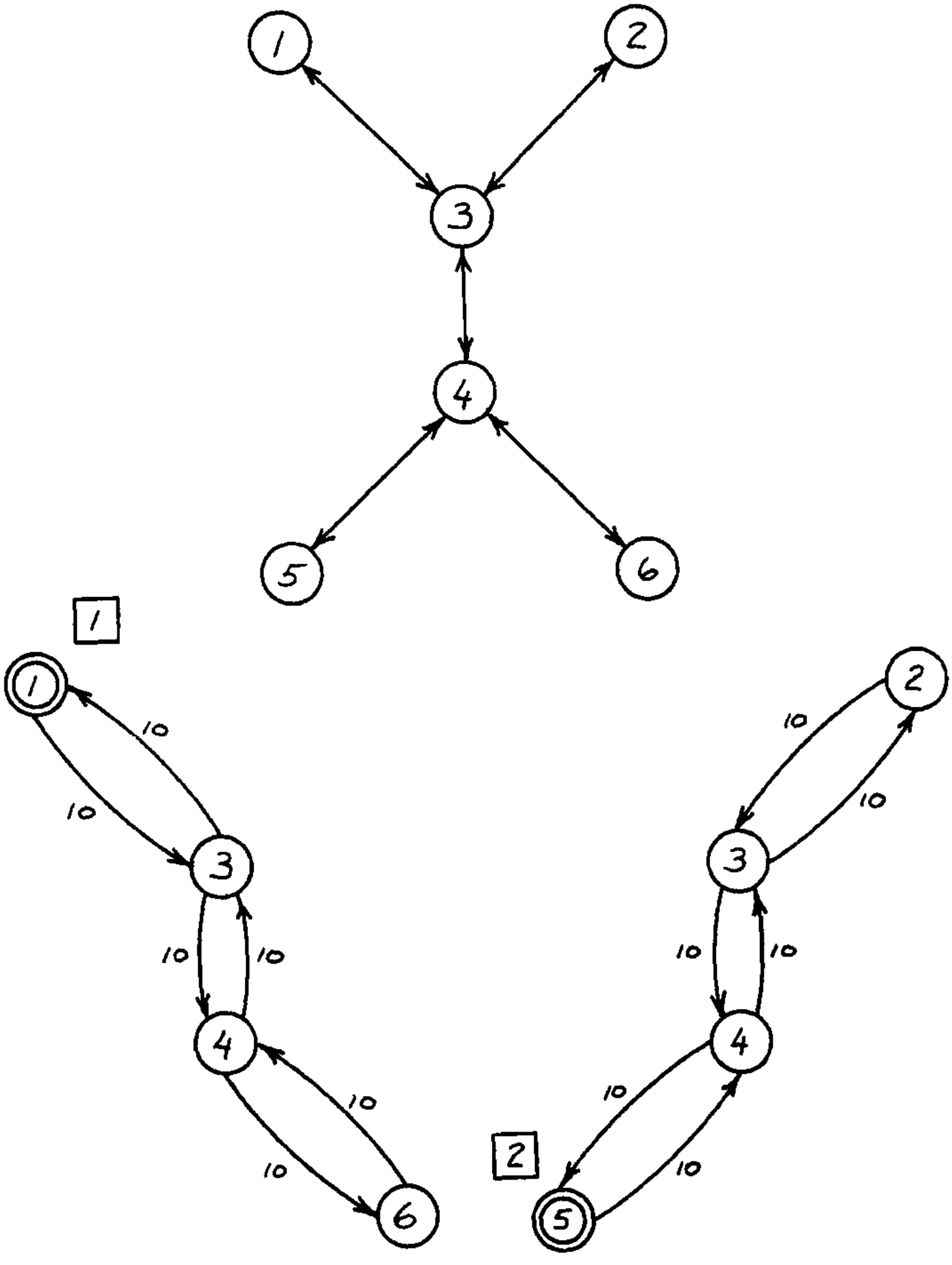


Figure 26. Problem E1 -- Final Set of Open Guideways and Final Routes and Service Frequencies

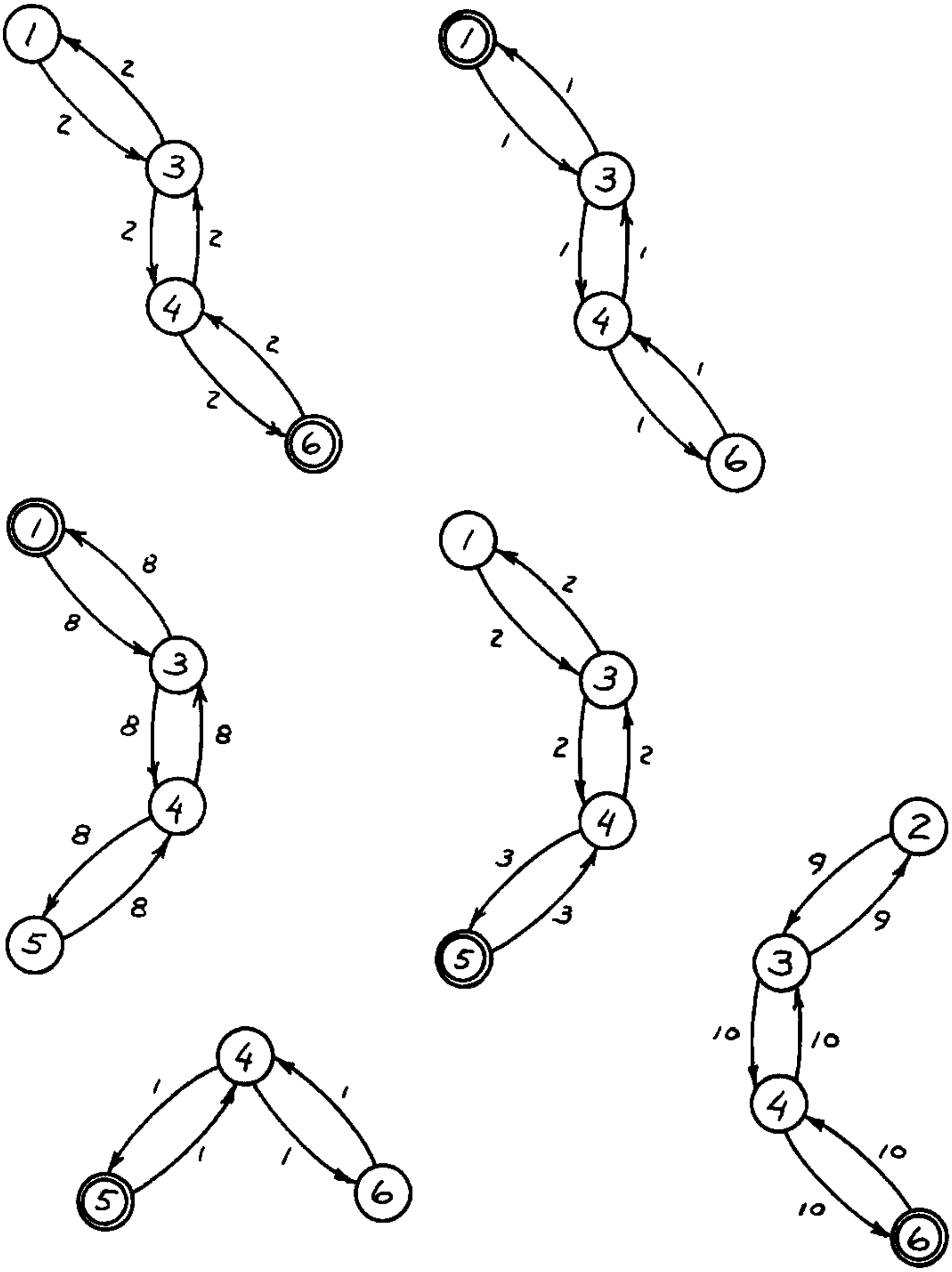


Figure 27. Test Problem E1 -- Sharp's Final Set of Routes and Vehicle Service Frequencies

Table 2. Comparison of Sharp's Guideway Algorithm to Multiple Guideway Algorithm for Problem E, Starting with E1

Sharp's Guideway Algorithm, Results of Problem E, Starting with E1

Ite.	Change	Passenger Time Costs	Vehicle Operating Costs	Guideway Fixed Costs	Total	Impvt.
0	Start	17,735	5,434	4,550	27,719	--
1	Insert (3,4)	10,174	4,072	4,610	18,856	8,863
2	Insert (4,5)	10,174	4,072	4,690	18,936	- 80
3	Delete (2,6)	9,969	2,976	3,110	16,055	2,801
4	Delete (1,5)	10,052	2,966	1,710	14,728	1,327
5	Insert (4,5)	6,172	2,606	1,790	10,568	4,160
6	Insert (2,3)	4,549	2,354	1,890	8,793	1,775
7	Insert (1,5)	4,549	2,354	3,290	10,193	-1,400
8	Delete (5,6)	4,444	2,098	1,090	7,632	1,161
9	Delete (1,2)	4,392	1,394	490	6,276	1,356
	Stop					

Final total costs = 6,276

Computation time = 217 seconds

Table 2. Continued

Multiple Guideway Algorithm, Results of Problem E, Starting with E1

Ite.	Change	Passenger Time Costs	Vehicle Operating Costs	Guideway Fixed Costs	Total	Impvt.
0	Start	17,721	5,134	4,550	27,405	--
1	Insert (3,4)	9,942	2,988	4,610	17,540	9,865
2	Delete (2,6)	9,590	2,872	3,110	15,572	11,833
3	Insert (4,5)	9,579	2,886	3,190	15,655	- 83
4	Delete (1,5)	7,078	1,892	1,790	10,760	4,812
5	Insert (2,3)	4,390	1,236	1,890	7,516	3,244
6	Delete (5,6)	4,566	1,182	1,090	6,836	680
7	Delete (1,2)	4,377	1,126	490	5,993	843
	Stop					

Final total costs = 5,993

Computation time = $156\frac{1}{2}$ seconds time of each run was approximately one minute longer for the runs using the multiple guideway algorithm. This increase in time resulted from one additional guideway algorithm iteration for each problem. It should be noted that both of these problems obtained their best solutions without inserting or deleting any guideways, because the initial set of open guideways was apparently optimal.

Test Problem D

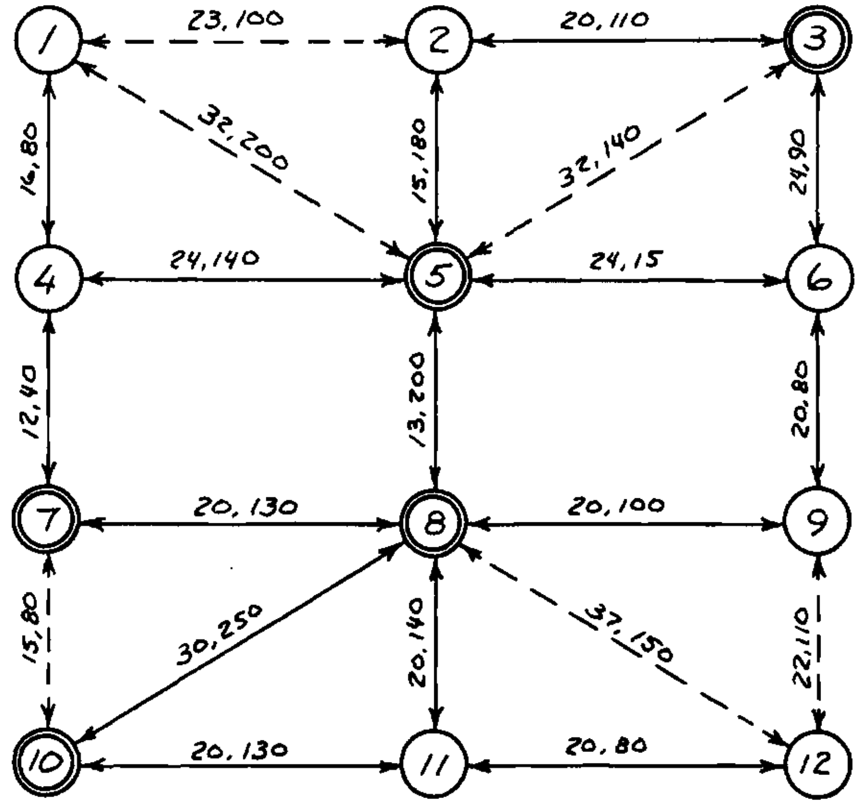
One subproblem of test problem D, D1a, was run using the multiple guideway algorithm. Due to the structure of the guideways and the travel demands, the multicommodity assignment algorithm required a high computation time. The program did result in lower costs than the run without the multiple guideway algorithm, but the problem was not run to completion due to the high execution time.

Route Restructuring

The route restructuring algorithm is demonstrated on test problems C1b and D1a. Route restructuring iterations are processed internal to the route algorithm iterations and costs are not computed if there is a definite cost decrease. The route algorithm is imbedded in Sharp's one-at-a-time guideway insertion-deletion algorithm.

Test Problem C1b

The initial set of open guideways is given in Figure 28, the initial set of routes and vehicle service frequencies is given in Figure 29, and the travel demands are presented in Table 3. Sharp's solution to this problem is presented in Figure 30. As can be seen, changes were few, limited to appending three guideways. The final routes are nine two-arc



Open guideway in problem C1

Closed guideway in problem C1

23,100 Travel time in minutes, relative fixed cost

Station location in problem C

Major station location in problem C

Figure 28. Station Locations and Possible Guideways in Problem C -- Open Guideways for Problem C1

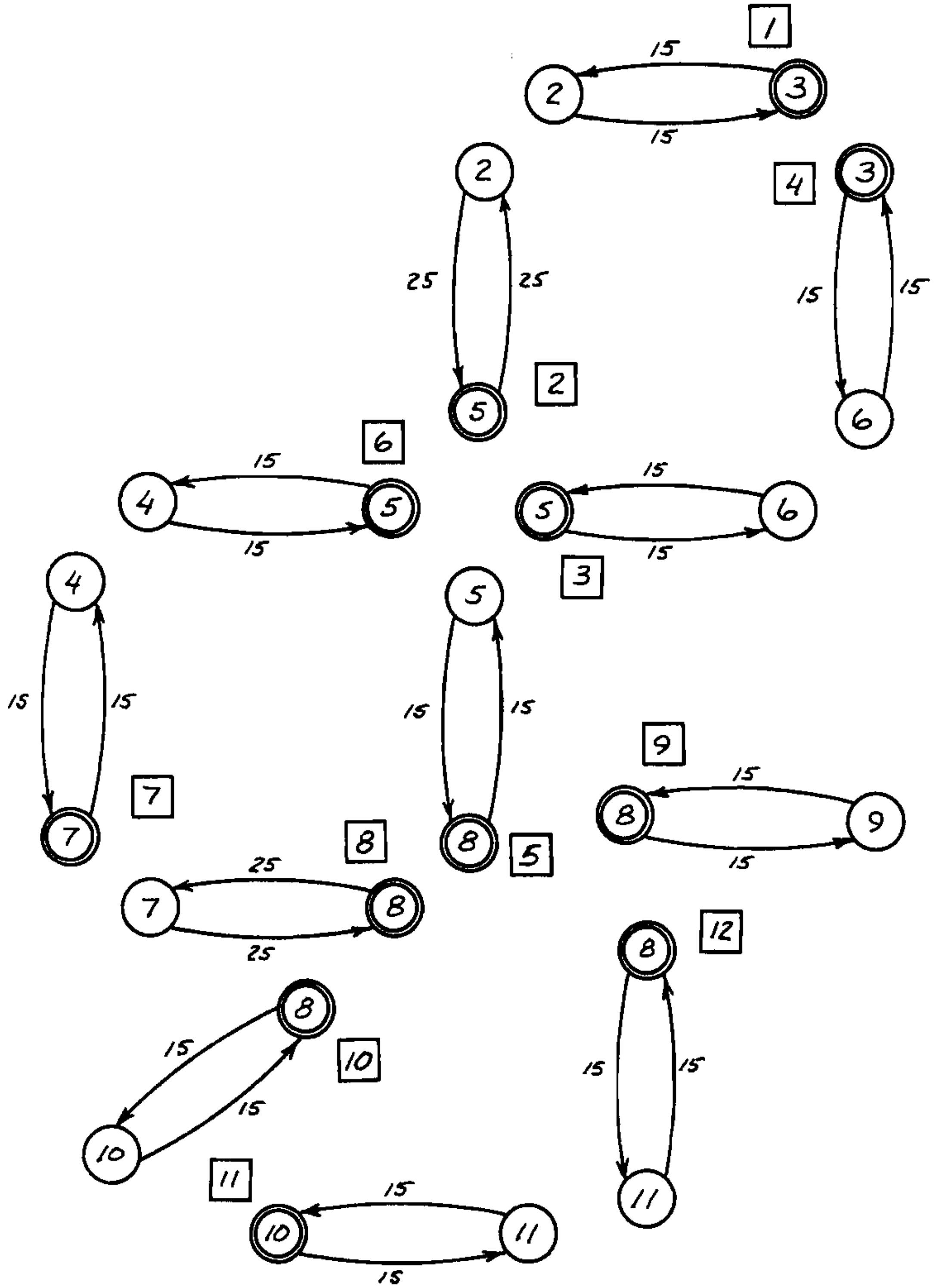


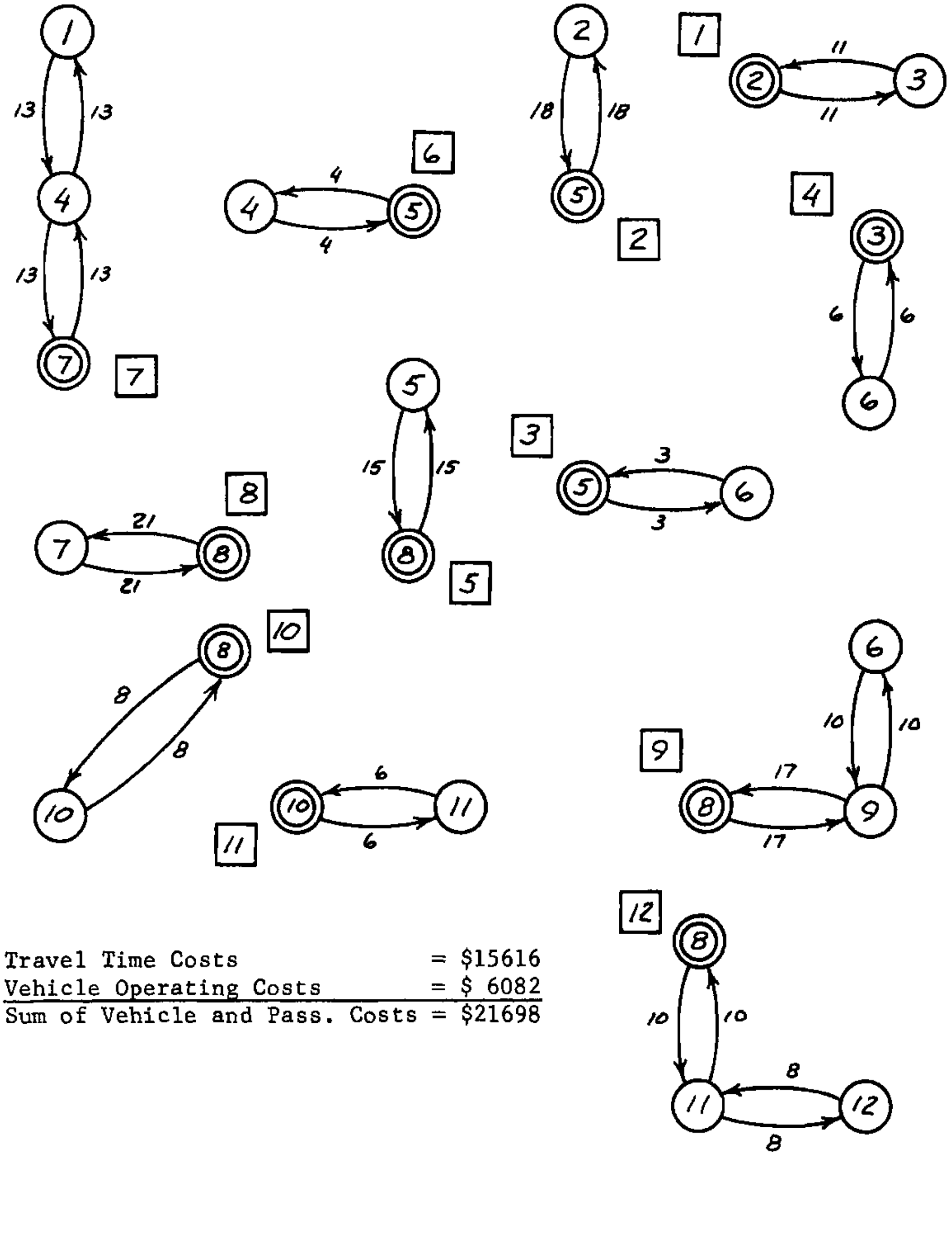
Figure 29. Problem C1, Starting with C1b, Initial Routes and Vehicle Service Frequencies

Table 3. Travel Demands in Problem C

Commodity Number	Origin Station	Destination Station	Travel Demand/Hr
1	1	4	500
2	1	5	200
3	1	7	200
4	1	8	400
5	2	3	200
6	2	5	300
7	2	6	100
8	2	8	600
9	3	2	300
10	3	5	400
11	3	6	400
12	3	8	400
13	3	9	100
14	4	1	200
15	4	5	200
16	4	7	100
17	4	8	500
18	5	2	100
19	5	4	100
20	5	6	100
21	5	8	500
22	6	3	100
23	6	5	300
24	6	8	700
25	6	9	200
26	7	4	100
27	7	5	100
28	7	8	800
29	7	9	200
30	7	11	100
31	8	5	100
32	8	7	200
33	8	9	100
34	8	11	300
35	9	6	100
36	9	8	700
37	9	11	300
38	10	5	100
39	10	8	700

Table 3. Continued

Commodity Number	Origin Station	Destination Station	Travel Demand/Hr
40	10	11	600
41	11	8	500
42	11	10	300
43	11	12	100
44	12	8	500
45	12	10	100
46	12	11	200



Travel Time Costs = \$15616
Vehicle Operating Costs = \$ 6082
Sum of Vehicle and Pass. Costs = \$21698

Figure 30. Solution for Test Problem C1b at the End of Guideway Algorithm Iteration 0 Using Sharp's Program

reversal routes, and three four-arc reversal routes. These routes result in high costs due to transfer times and are very unattractive to potential passengers. This solution is a prime example of why route restructuring is needed -- passengers are not going to utilize a transit system if they can travel only a distance of one or two stations on any route.

The passenger flow assignments on the initial routes lead to an infeasible solution. This is due mainly to no routes covering station one and station twelve. In route algorithm iteration zero, the following routes are combined:

route one with route two,

route six with route five,

route three with route four,

routes five and six to route seven, and

route eleven to route twelve.

The routes formed by the addition of route eleven and route twelve are now reversed and the frequency of service between station eight and station eleven is reduced to one vehicle per time period. The route constructed by combining routes three and four is also reversed with service between stations five and six increased to nineteen vehicles and service between stations six and three decreased to one vehicle. These new routes are given in Figure 31 and the costs of these routes are:

Travel Time Costs	= \$18109
Vehicle Operating Costs	= \$ 7320
Sum of Vehicle and Passenger Costs	= \$25429

The route algorithm now appends the following guideways:

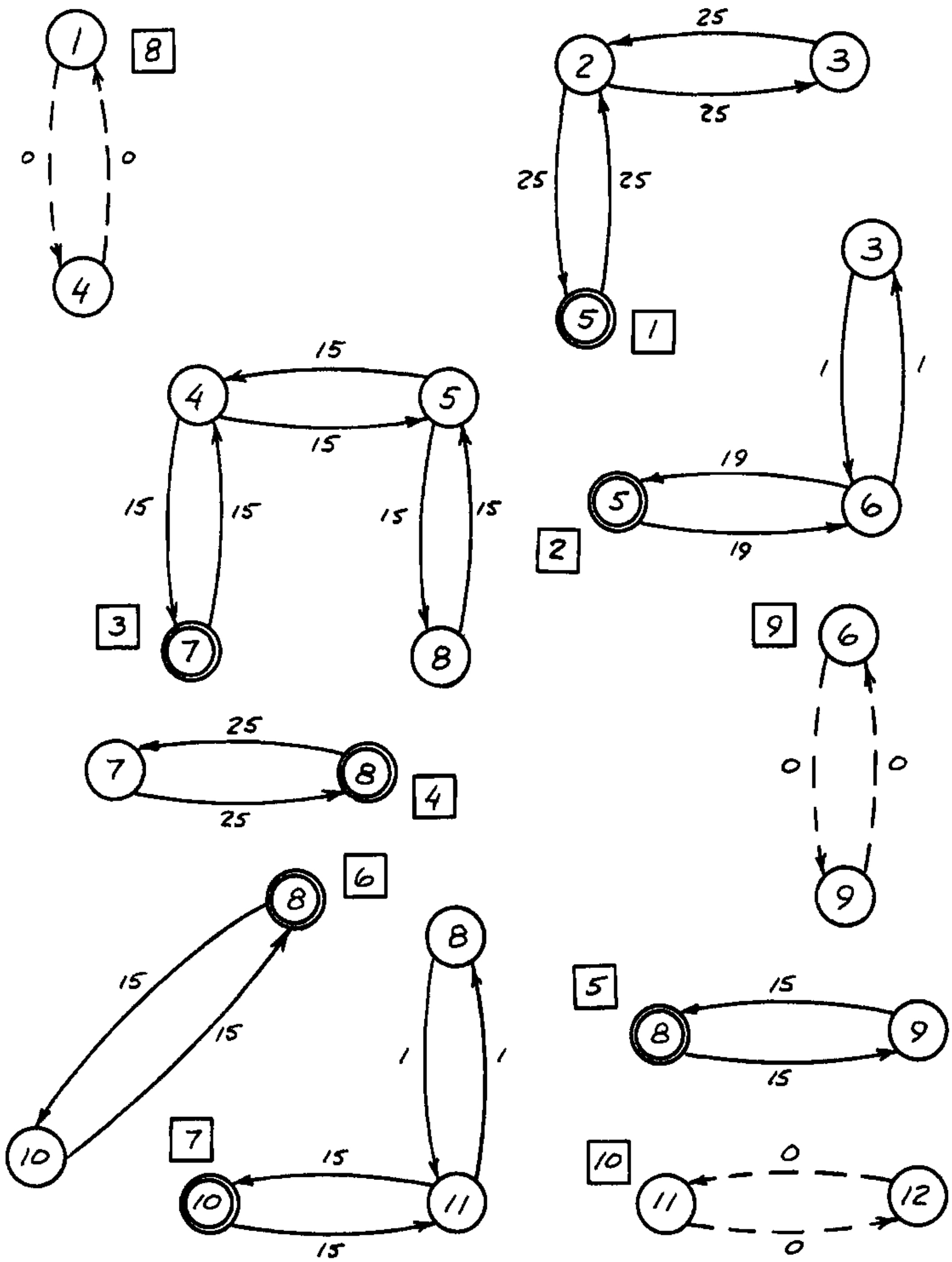


Figure 31. Test Problem C1b -- Routes and Vehicle Service Frequencies at the End of Route Algorithm Number 0 During Guideway Algorithm Number 0

guideway (5,8) to route seven,

guideway (8,11) to route three,

guideway (11,12) to route three,

guideway (6,9) to route five,

guideway (3,6) to route five, and

guideway (3,6) to route one

to result in the set of routes shown in Figure 32. The costs of these routes are:

Travel Time Costs	= \$16265
Vehicle Operating Costs	= \$ 8802
Sum of Vehicle and Passenger Costs	= \$25067

Service is now increased on routes one, three, and five to reduce the total of vehicles and passenger costs by \$1485 while moving towards feasibility. A new reversal route is constructed connecting stations five, eight, seven, four, and one, thus eliminating the zero level route between stations one and four. A vehicle service frequency of two vehicles per time period is given to the entire route. The route algorithm now appends the following set of guideways:

(4,7) to route four, and

(1,4) to route four,

and proceeds to increase service on routes four and eight. At this point the costs of the routes are:

Travel Time Costs	= \$14059
Vehicle Operating Costs	= \$10720
Sum of Vehicle and Passenger Costs	= \$24779

The route restructuring algorithm is now entered again since the route algorithm is entering the phase for decreasing service on existing

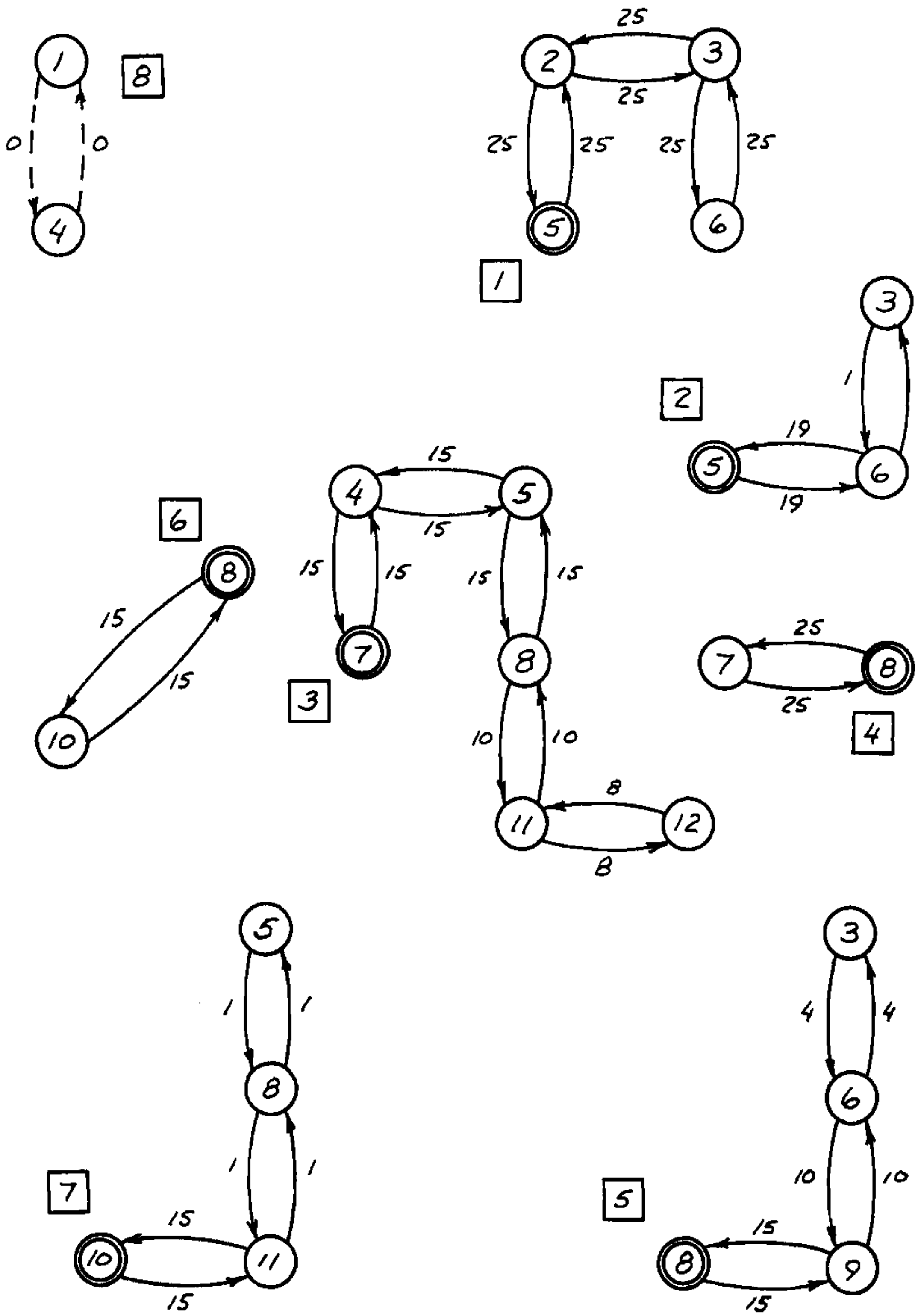


Figure 32. Test Problem Clb -- Routes and Vehicle Service Frequencies after Route Algorithm Number 6

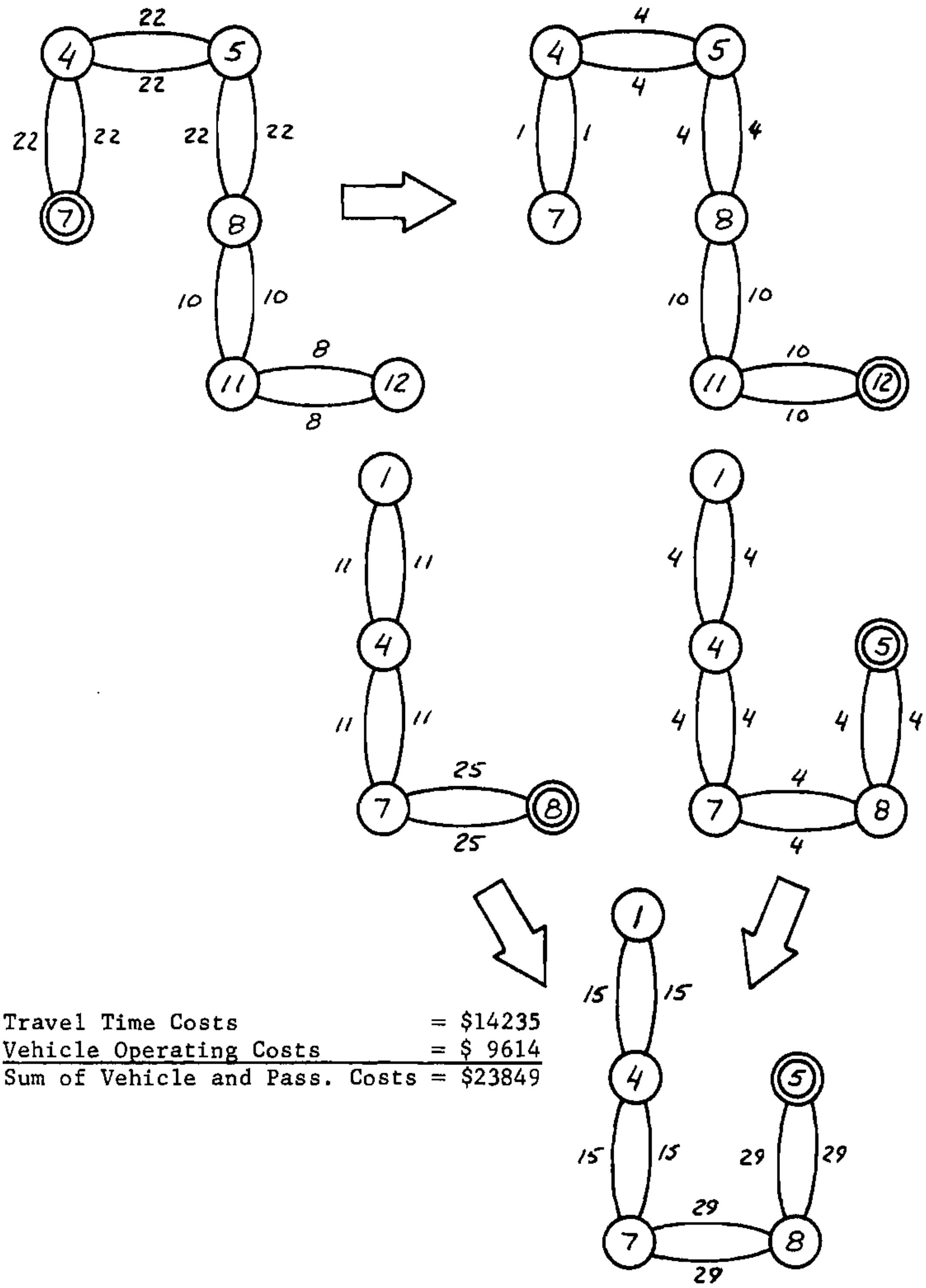
routes, and two route changes are made. Route three is reversed resulting in a net decrease of one hundred and ten vehicles per time period. This will increase the passenger waiting time but will decrease vehicle operating costs. The next change is to add route four to route eight. This change is checked for cost since the first arc of route eight will have its service increased by twenty-five vehicles per time period. The costs do decrease and the combination is made. The resulting routes and costs are given in Figure 33.

The route algorithm now decreases excess capacity on routes, increases capacity on routes, attempts but fails to construct a new route, and attempts and fails to decrease more capacity on existing routes. The route restructuring algorithm cannot manipulate the routes further, for lack of cost decreases, and thus the route algorithm is exited. The final routes and costs are given in Figure 34 and Sharp's final routes and costs are given in Figure 30. This program achieved a reduction of five in the number of routes and a decrease in overall costs of $273, or 5% of passenger time costs, compared to Sharp's program.

Test Problem D1a

Test problem D is the largest of the three problems, having the same number of stations as problem C, twelve, but having twenty-five possible guideways compared to twenty-one for problem C. The initial set of open guideways is given in Figure 35, the initial set of routes and service frequencies is given in Figure 36, and the travel demands are given in Table 4.

The program uses the initial set of open guideways to enter the route scheduling algorithm. Flow assignments are computed first; then



Travel Time Costs	= \$14235
Vehicle Operating Costs	= \$ 9614
Sum of Vehicle and Pass. Costs	= \$23849

Figure 33. Test Problem C1b -- Route Restructuring Algorithm

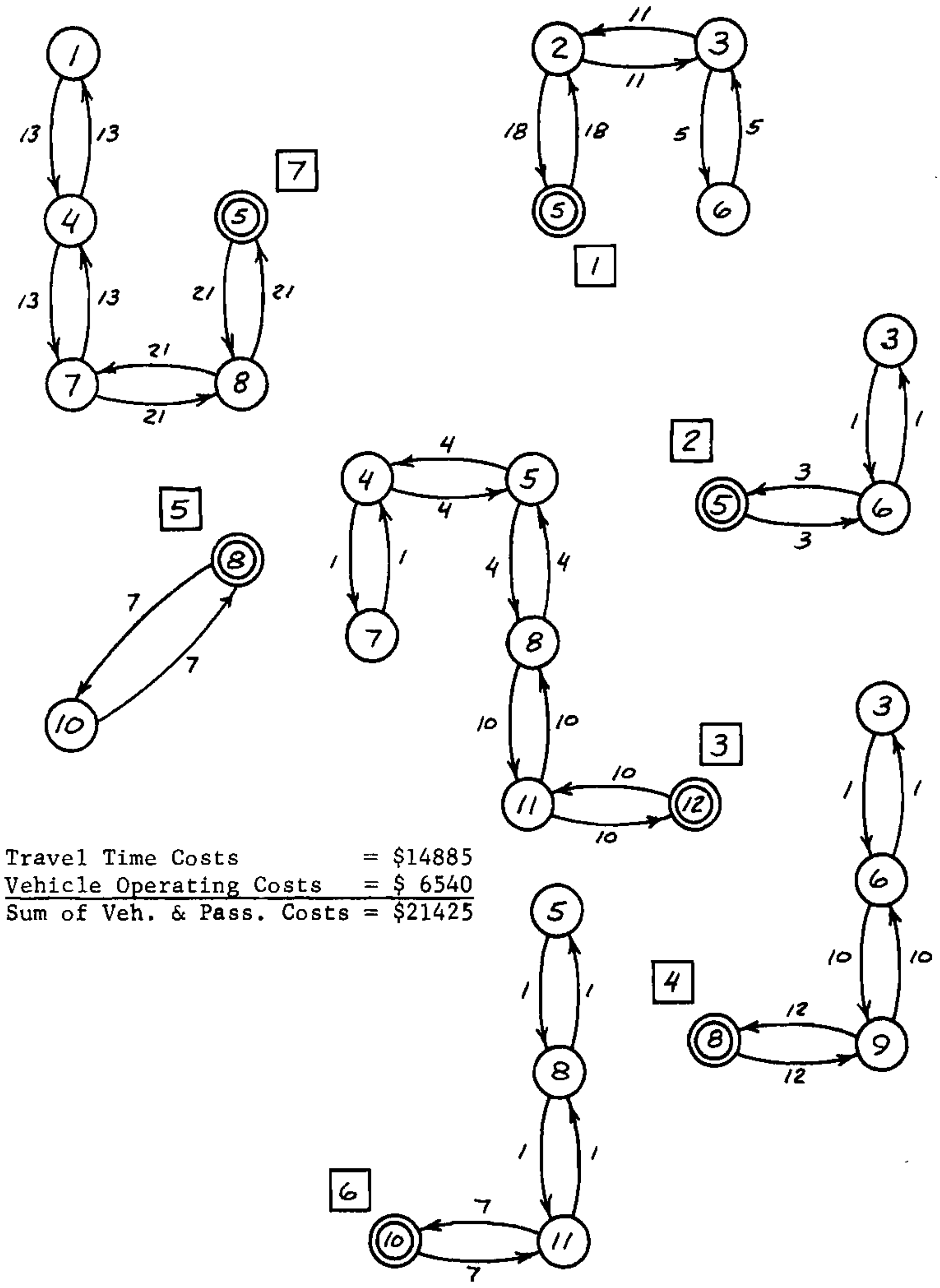
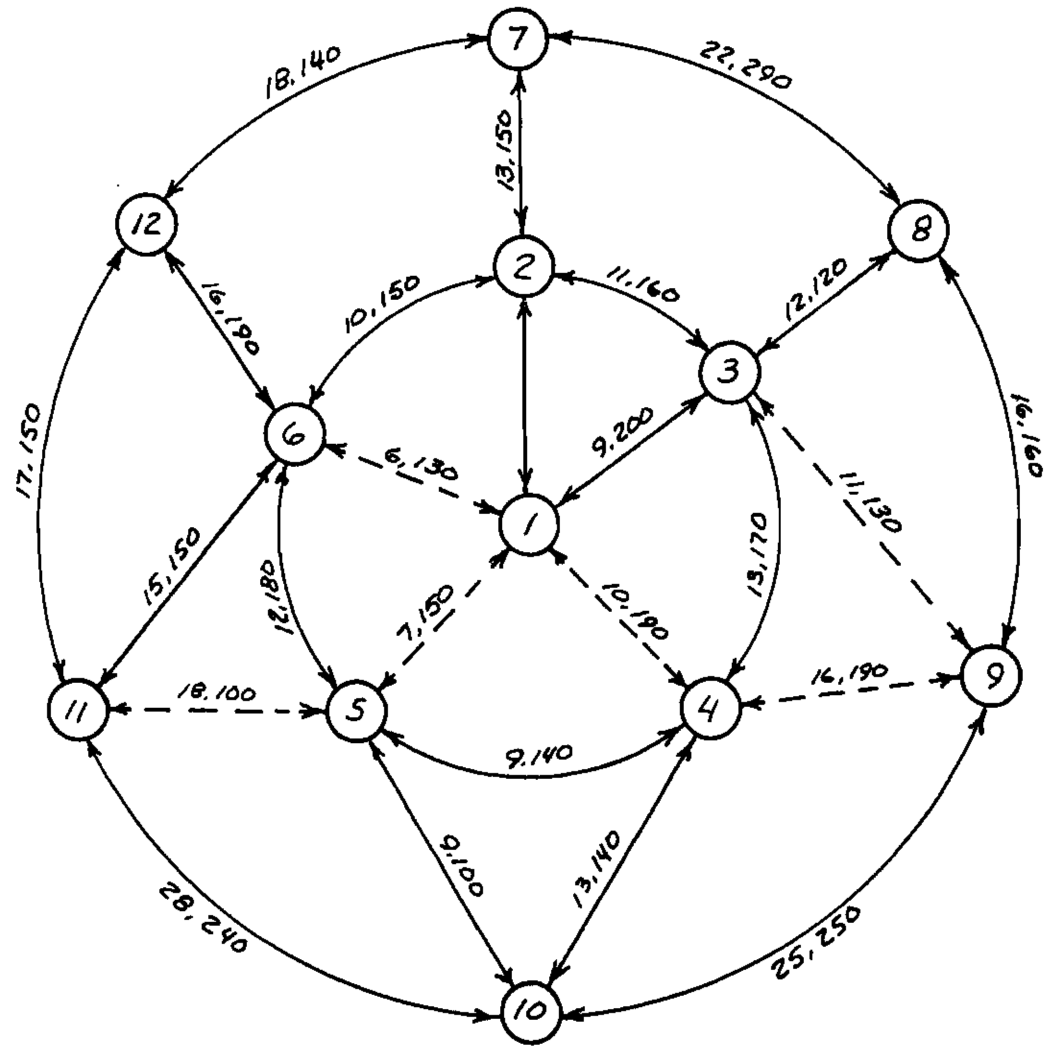


Figure 34. Test Problem C1b at the End of Guideway Algorithm Iteration 0



Open Guideways in Problem D1

Closed Guideways in Problem D1

18,140 Travel Time in Minutes, Relative Fixed Costs

Figure 35. Station Location and Possible Guideways for Test Problem D -- Open Guideways for Problem D1

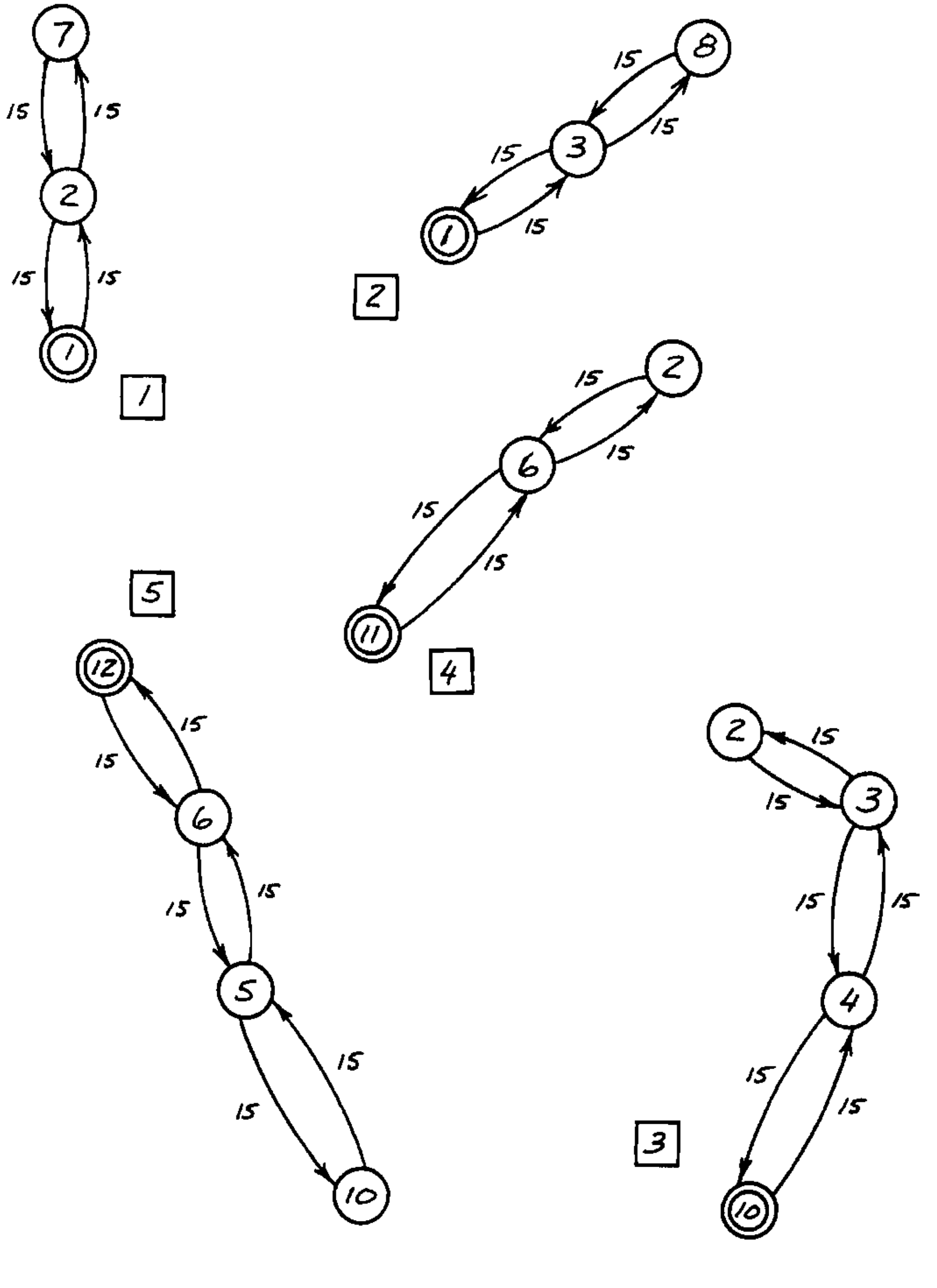


Figure 36. Initial Routes and Vehicle Service Frequencies for Test Problem D1

Table 4. Problem D, Travel Demands

Commodity Number	Origin Station	Destination Station	Travel Demand
1	1	2	300
2	1	8	200
3	1	10	200
4	1	11	200
5	2	1	300
6	2	4	200
7	2	5	200
8	2	12	200
9	3	5	300
10	3	6	300
11	3	10	200
12	3	11	200
13	4	2	300
14	4	6	300
15	4	7	200
16	4	8	200
17	5	2	300
18	5	3	300
19	5	9	200
20	5	12	200
21	6	3	300
22	6	4	400
23	6	8	200
24	6	9	200
25	7	1	300
26	7	3	300
27	7	6	300
28	7	9	200
29	7	11	200
30	8	6	200
31	8	2	300
32	8	10	200
33	9	12	200
34	9	1	200
35	9	2	300
36	9	7	200
37	9	11	100
38	10	1	300
39	10	3	300
40	10	6	300

Table 4. Continued

Commodity Number	Origin Station	Destination Station	Travel Demand
41	10	8	200
42	10	12	200
43	11	1	200
44	11	4	300
45	11	7	200
46	11	9	100
47	12	3	300
48	12	4	300
49	12	8	200
50	12	10	200

the route restructuring algorithm is entered and two changes are made. Route three, a reversal route covering stations ten, four, three, and two is reversed so that it originates at station two instead of station ten. It is now added to the end of route four which covered stations eleven, six, and two, so that the newly formed route now starts at station eleven and proceeds to stations six, two, three, four, and ten before looping back on itself. This change does not increase cost at any stage since there is no change in vehicle service to raise vehicle operating costs and to increase passenger delay costs. Savings are attributed to a decrease in transfer time for passengers using both routes three and four.

The new route four (old route five) is now reversed with the first two arcs having their vehicle service increased by two units, and the third arc's service decreased by ten units. Thus there is a reduction of twelve vehicles per time period which decreases the vehicle operating costs. Passenger costs increase due to longer waiting times for passengers boarding onto route four at station twelve or boarding route four at station six traveling to station twelve. Passenger costs are decreased by the reduction in waiting and delay time costs by passengers boarding route four at stations ten and five traveling in either direction and by passengers boarding route four at station six traveling towards station five. The net result is a decrease in total travel time costs of $1887, from $29899 to $28012, and in vehicle operating costs of $386, from $4230 to $3844. These changes are shown in Figure 37.

The route restructuring algorithm is not successfully used again until the third guideway algorithm, when it reverses a route. This change is not reflected in the final results since the outcome of the guideway

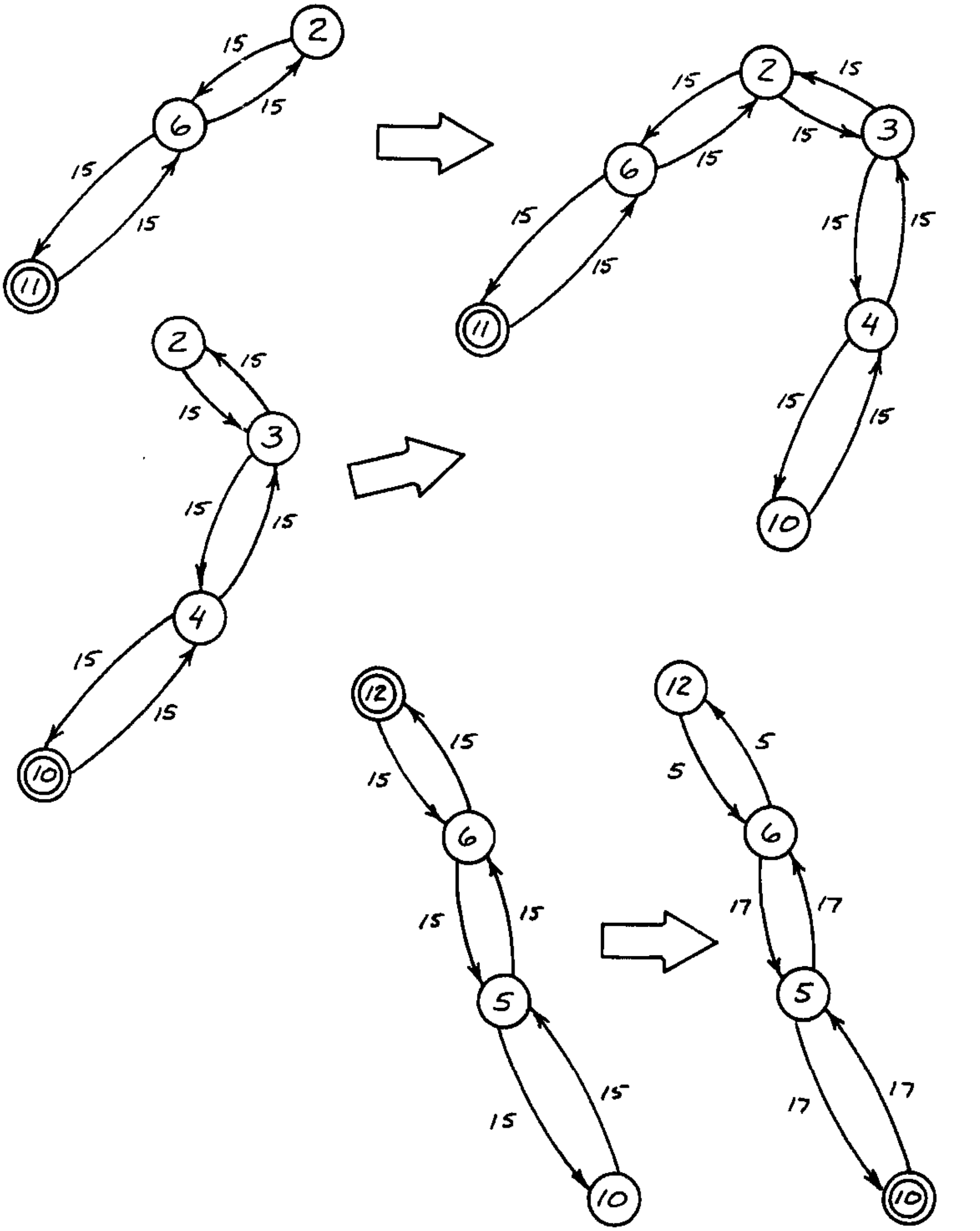


Figure 37. Test Problem D1 -- Results of Route Restructuring Algorithm

algorithm was negative. The final set of routes, vehicle service frequencies, and costs for this problem is given in Figure 38. The results are twenty open guideways for this program compared to sixteen open in Sharp's solution and a total cost reduction of $1197, or 7% of passenger time costs. The results obtained from Sharp's algorithm are given in Figure 39.

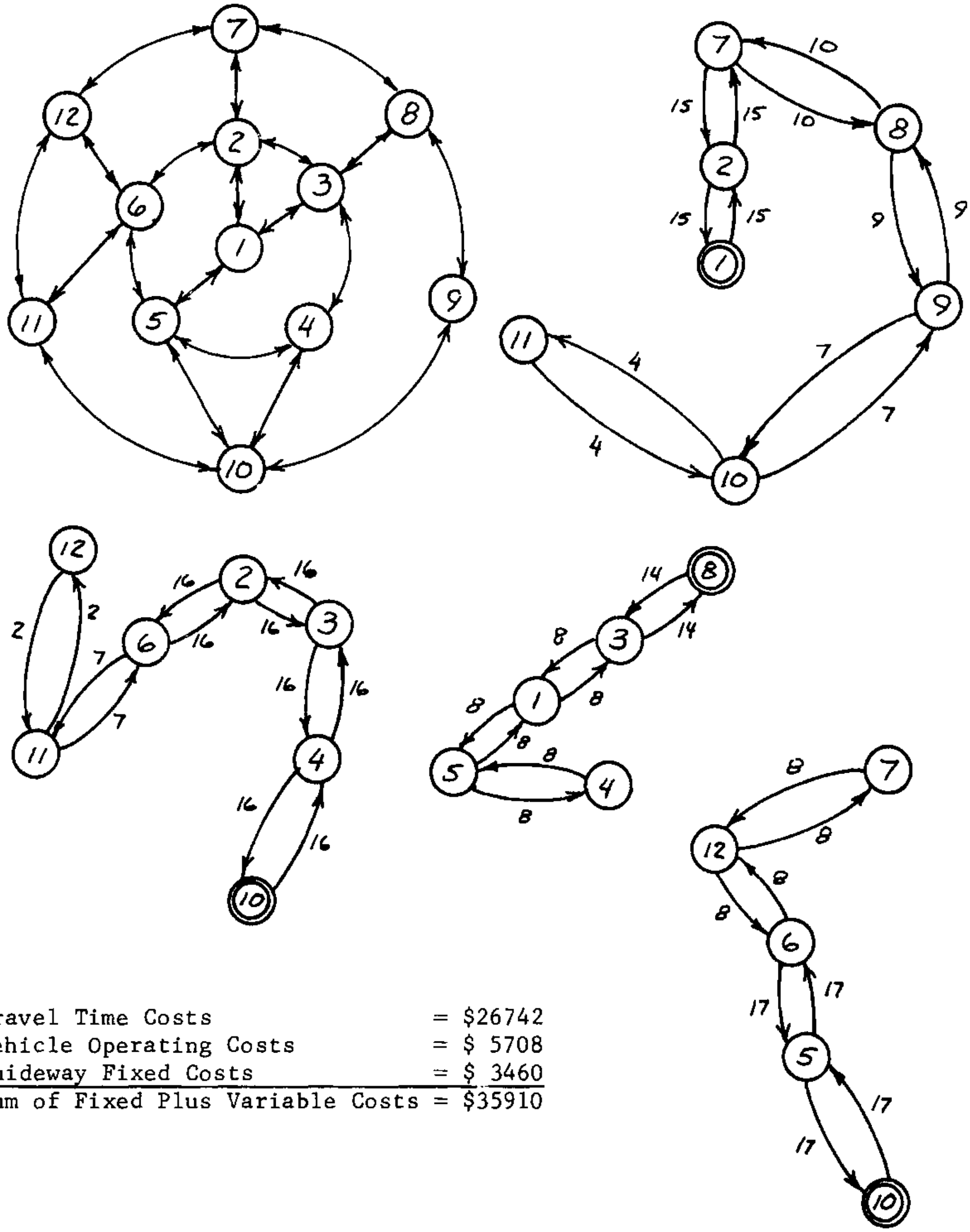
Test Problem C

Test problem C was solved using four different sets of initial open guideways, C1, C2, C3, and C4, which are given in Figure 40. Problem C1 was run using three extremely different sets of initial routes, C1a, C1b, and C1c, while problems C2, C3, and C4 were run with only one initial set of routes. These initial routes are given in Figures 41, 42, 43, 44, and 45, with the initial routes for problem C1b already shown in Figure 29.

Problem C1a uses the same set of open guideways as previously described in problem C1b, but the routes are CBD-oriented. These routes are left intact with some appending and route combining performed. This problem resulted in the lowest costs obtained from the C problem.

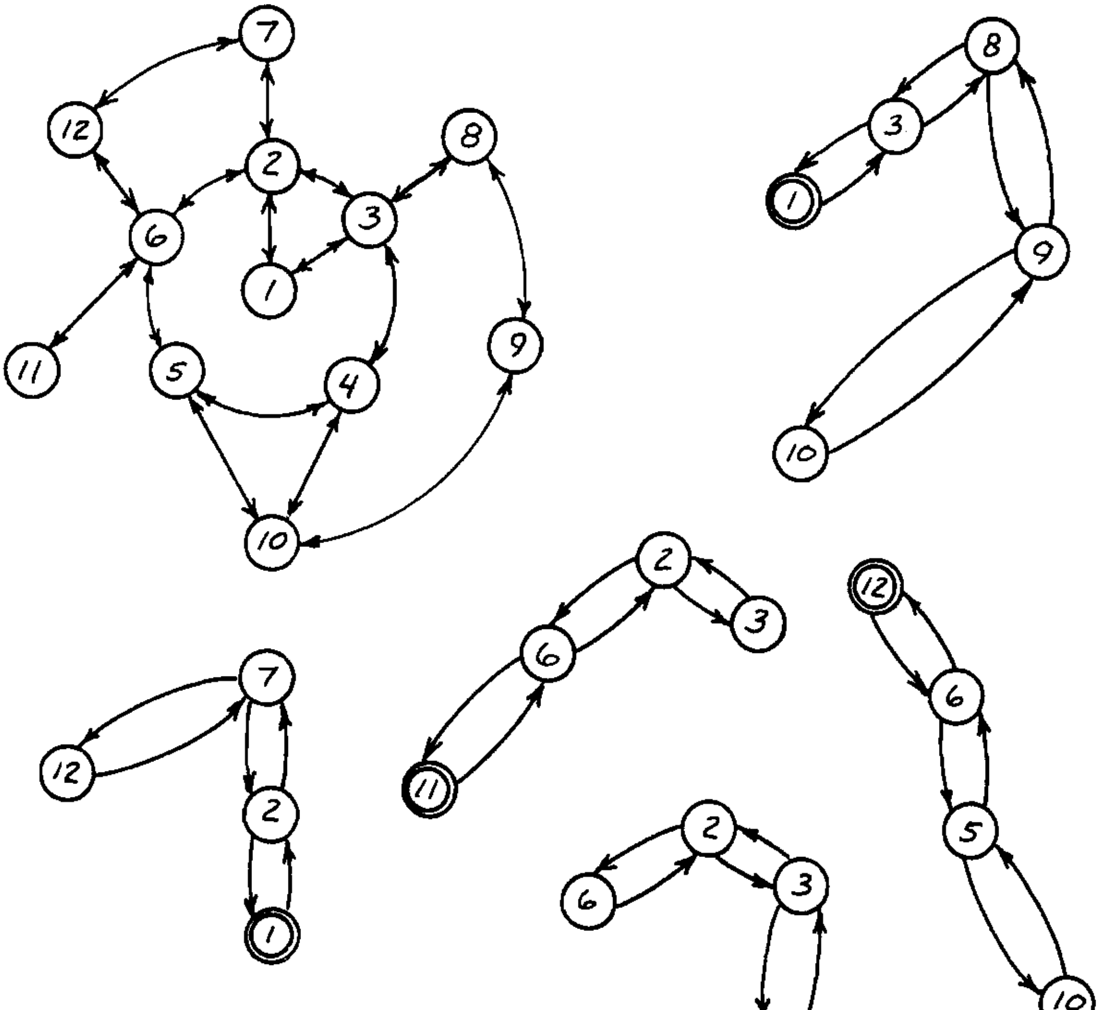
Problem C1b, which has been previously described earlier in this chapter, uses a different approach than problem C1a. It starts with all two-arc reversal routes. Sharp's program did little to this system except append a few arcs, but this program combined routes to result in CBD-oriented routes. Problem C1c started with just one two-arc reversal route and let the program design the routes needed. The results were four routes, not CBD-oriented, and about average costs.

Problems C2, C3, and C4 have exactly the same set of initial routes and service frequencies as problem C1a but with a variety of open guideways,



Travel Time Costs	= \$26742
Vehicle Operating Costs	= \$ 5708
Guideway Fixed Costs	= \$ 3460
Sum of Fixed Plus Variable Costs	= \$35910

Figure 38. Test Problem Dla -- Final Routes, Vehicle Service Frequencies, Open Guideways, and Costs



Travel Time Costs	= $28657
Vehicle Operating Costs	= $ 5820
Guideway Fixed Costs	= $ 2630
Sum of Fixed Plus Variable Costs	= $37107

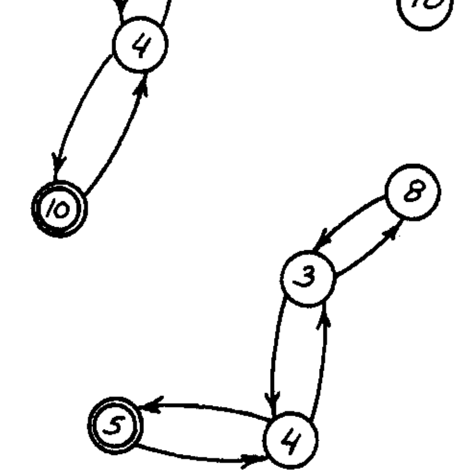
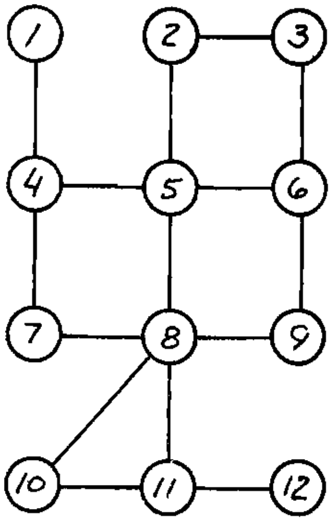
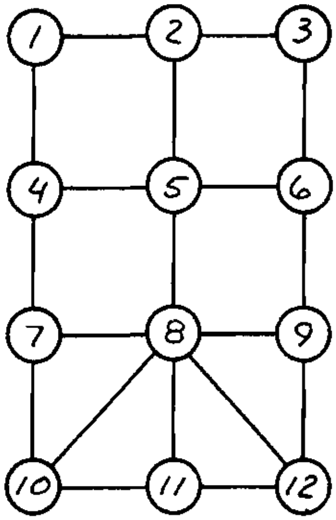


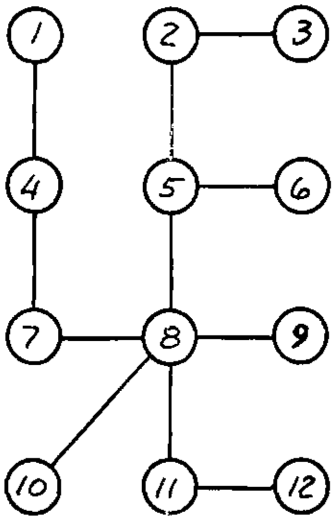
Figure 39. Test Problem D, Starting with D1 -- Sharp's Final Set of Open Guideways, Routes, and Costs



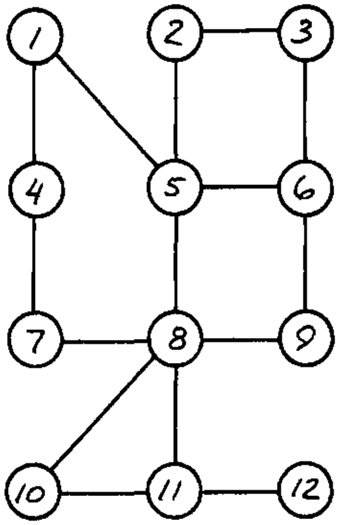
C1



C2



C3



C4

Figure 40. Problem C -- Initial Sets of Open Guideways

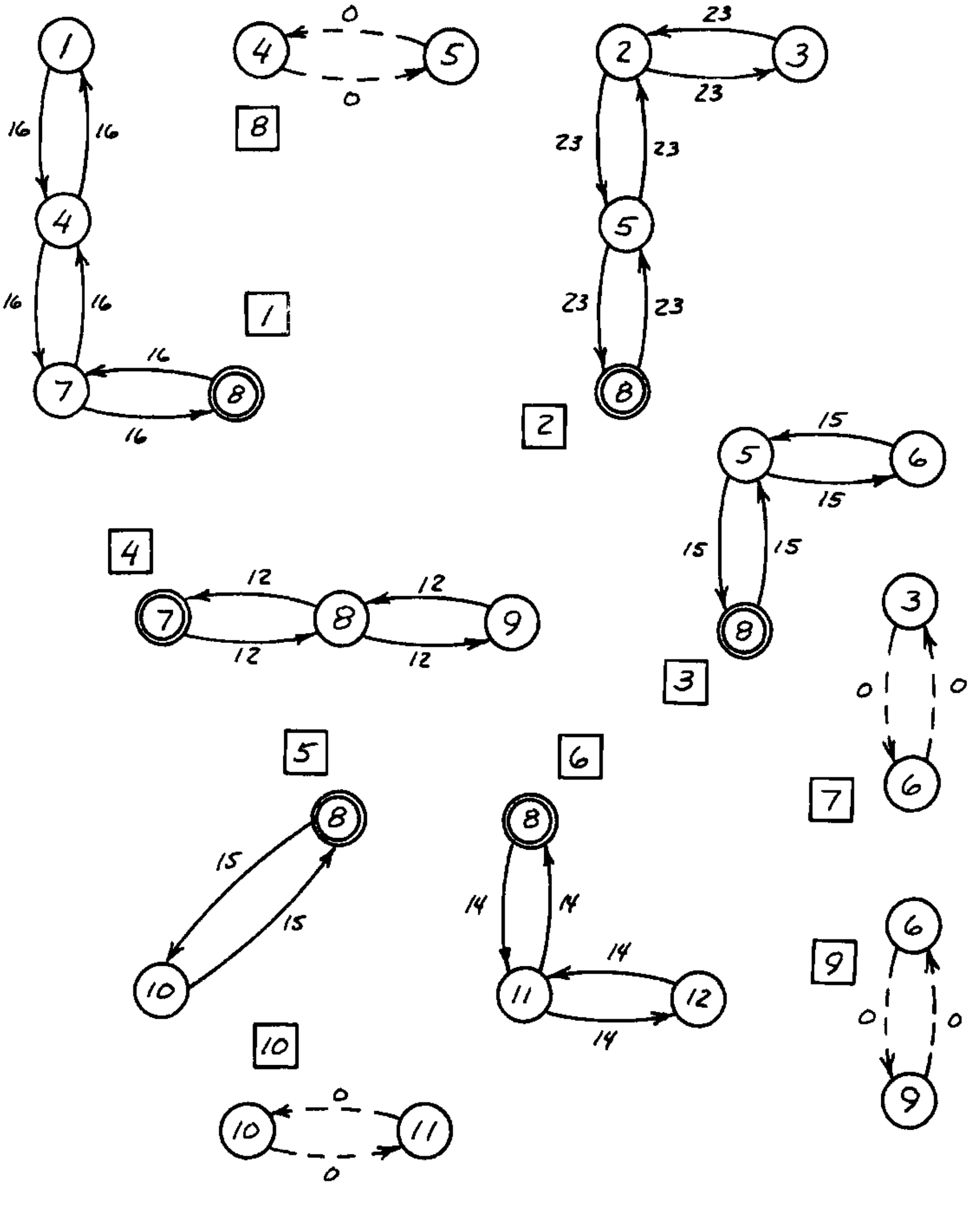


Figure 41. Test Problem C1, Starting with C1a -- Initial
Routes and Vehicle Service Frequencies

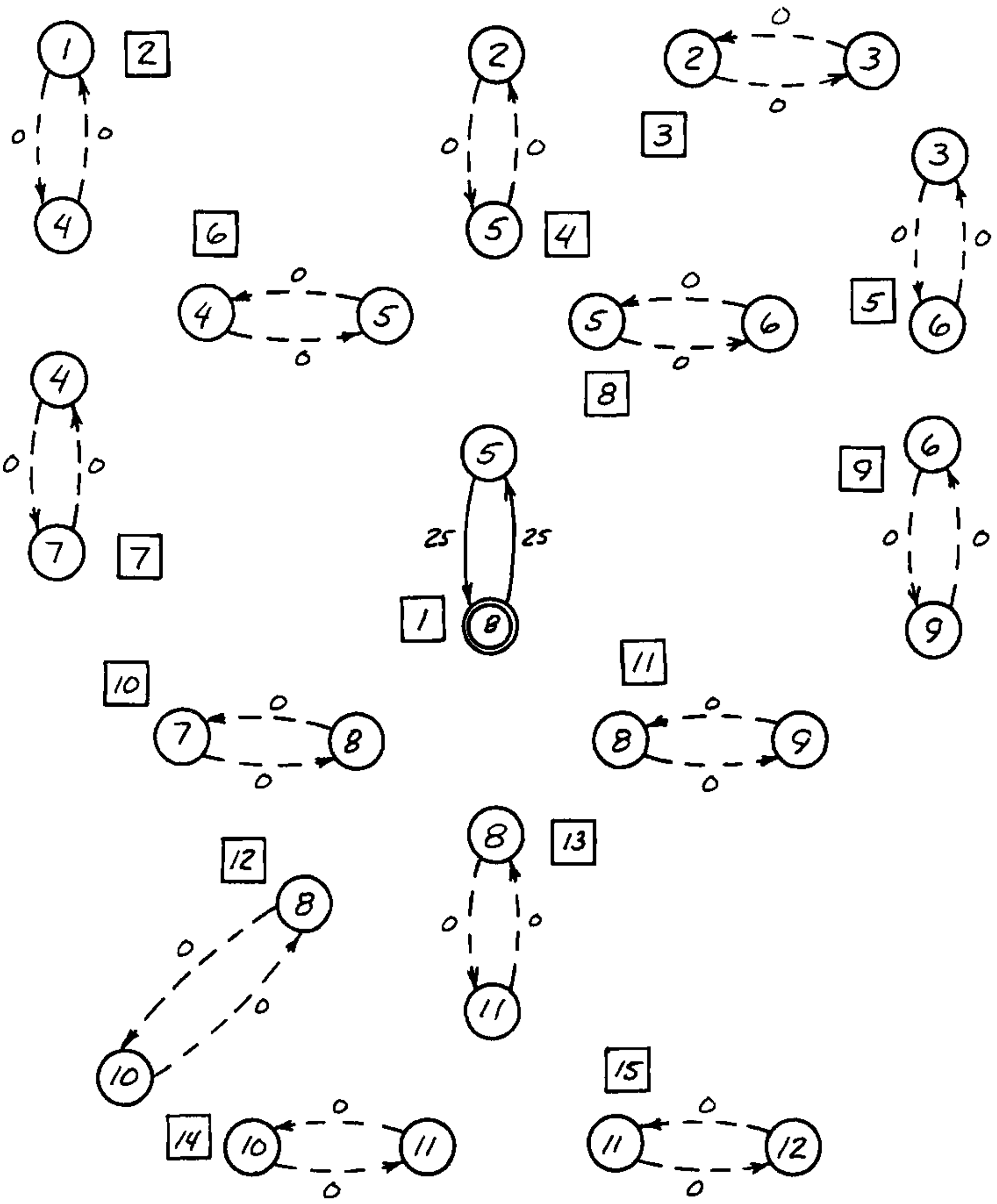


Figure 42. Test Problem C1, Starting with C1c -- Initial Routes and Vehicle Service Frequencies

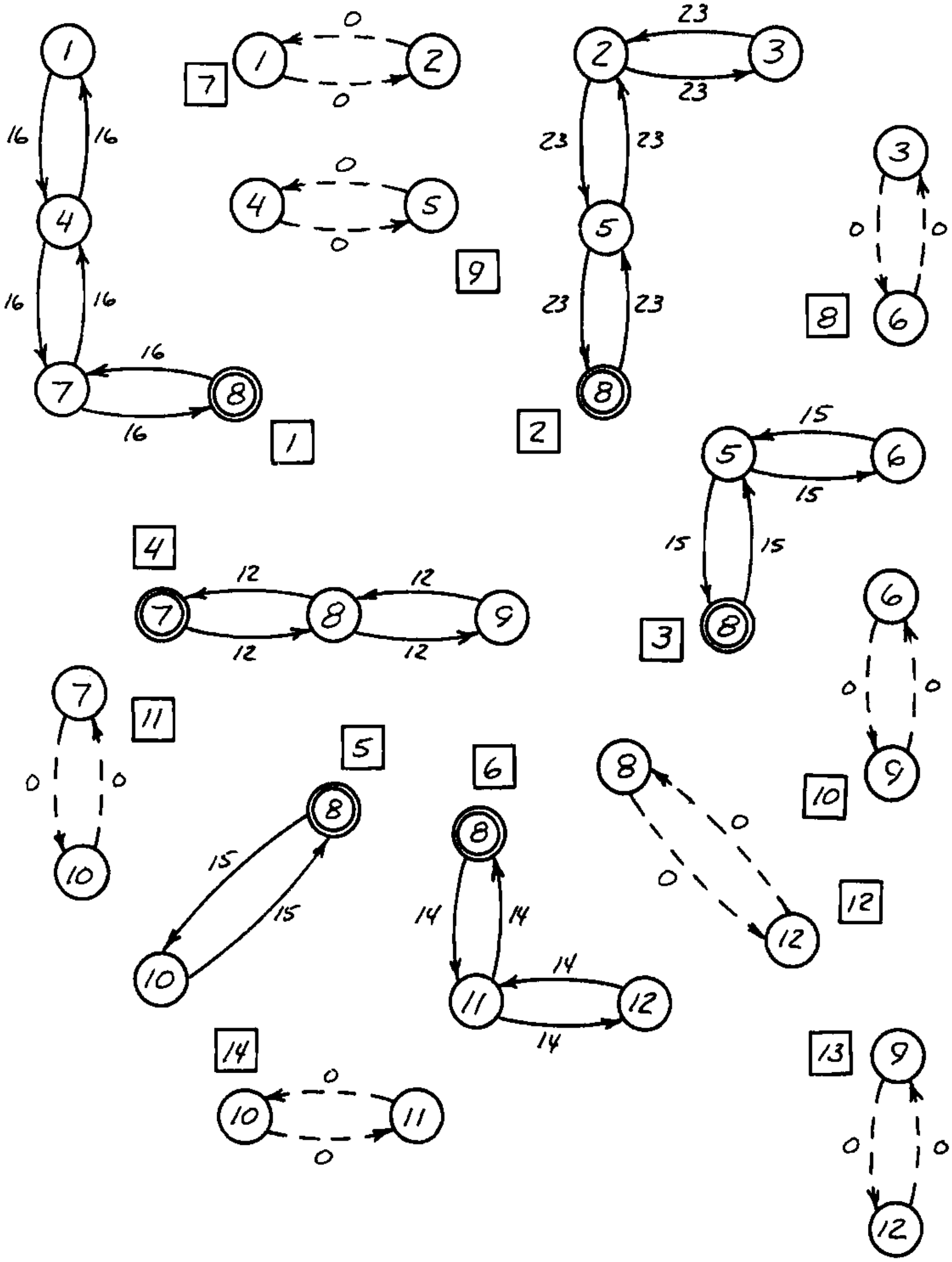


Figure 43. Test Problem C2 -- Initial Routes and Vehicle Service Frequencies

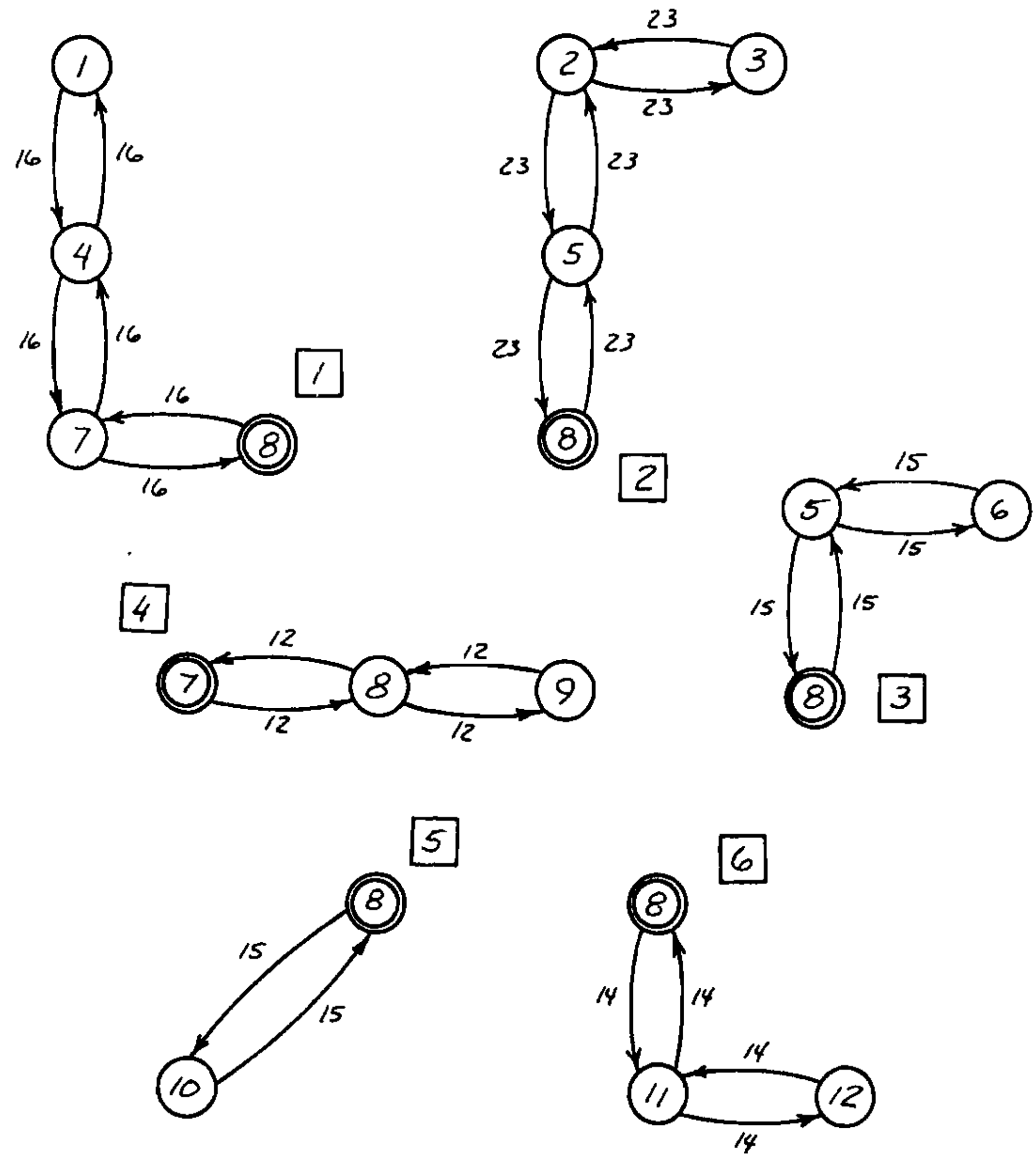


Figure 44. Test Problem C3 -- Initial Routes and Vehicle Service Frequencies

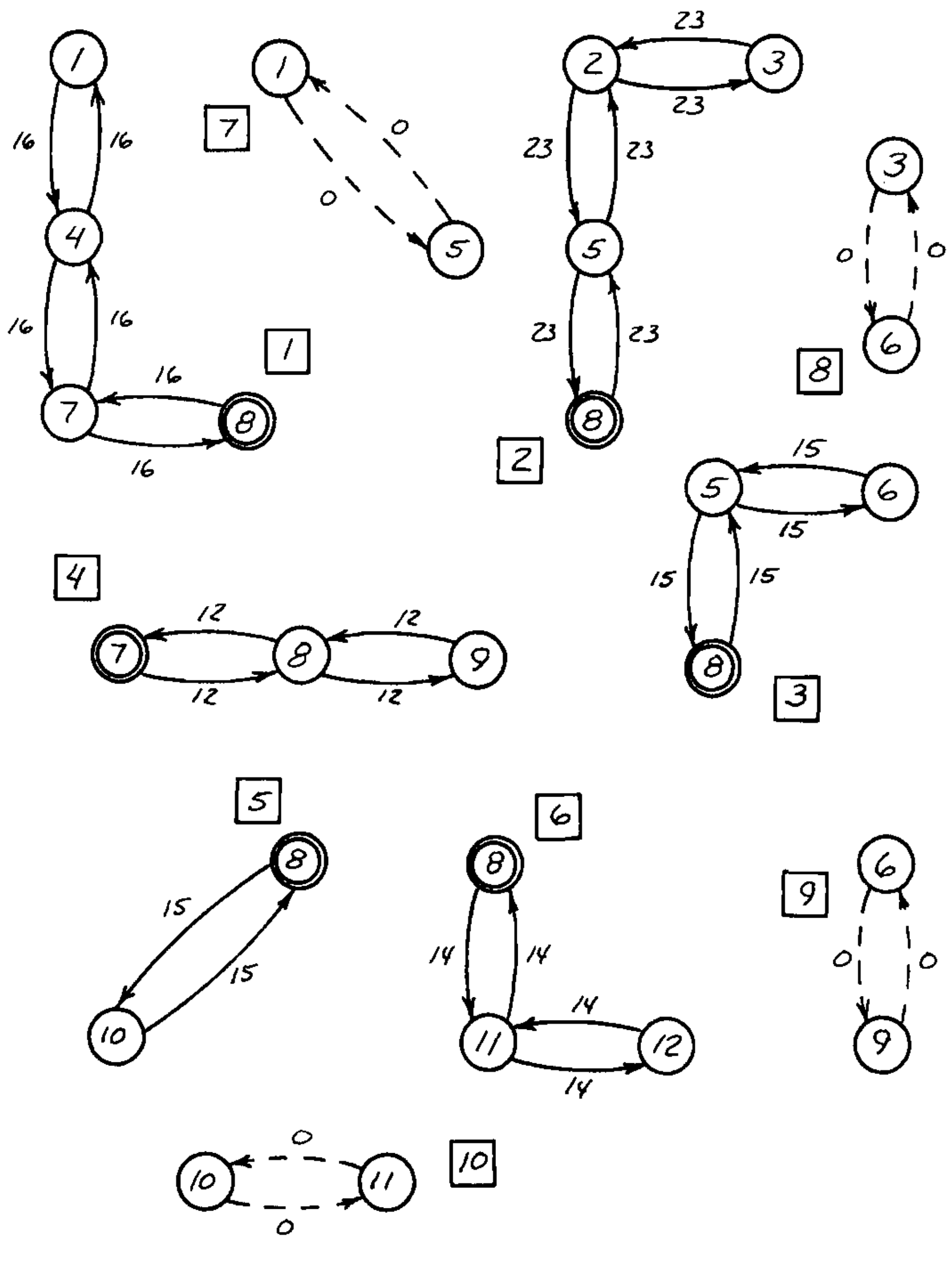


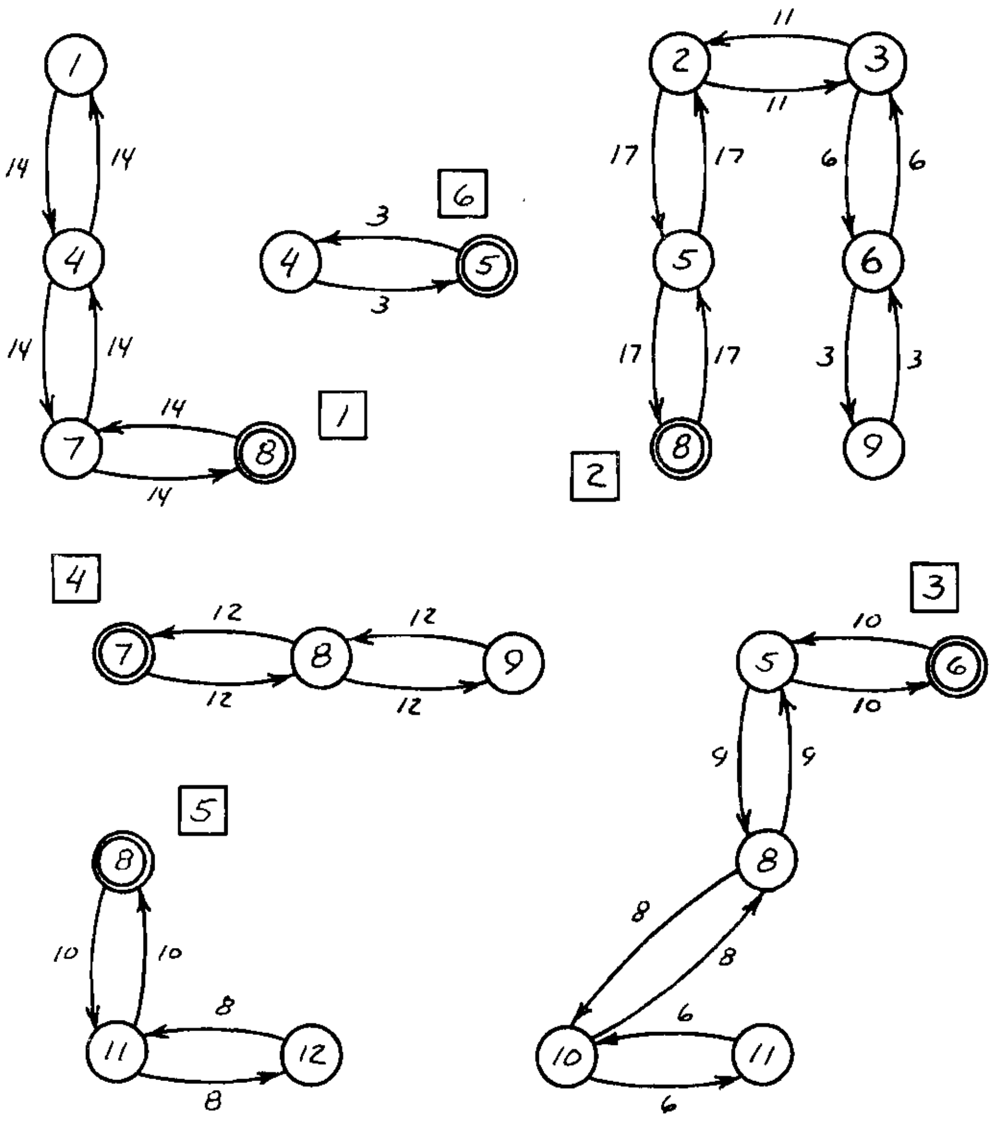
Figure 45. Test Problem C4 -- Initial Routes and Vehicle Service Frequencies

from nineteen open in C2 to eleven open in C3. Problem C4 is the same as C1a with arc (1,5) substituted for arc (4,5). The final set of open guideways for C2 and C4 are the same with C3 having two less open guideways and slightly higher costs. The final results for these runs are given in Figures 46, 47, 48, 49, and 50. Figures 51 and 52 show the final sets of open guideways and total costs obtained from Sharp's program and this program, respectively.

Test Problem D

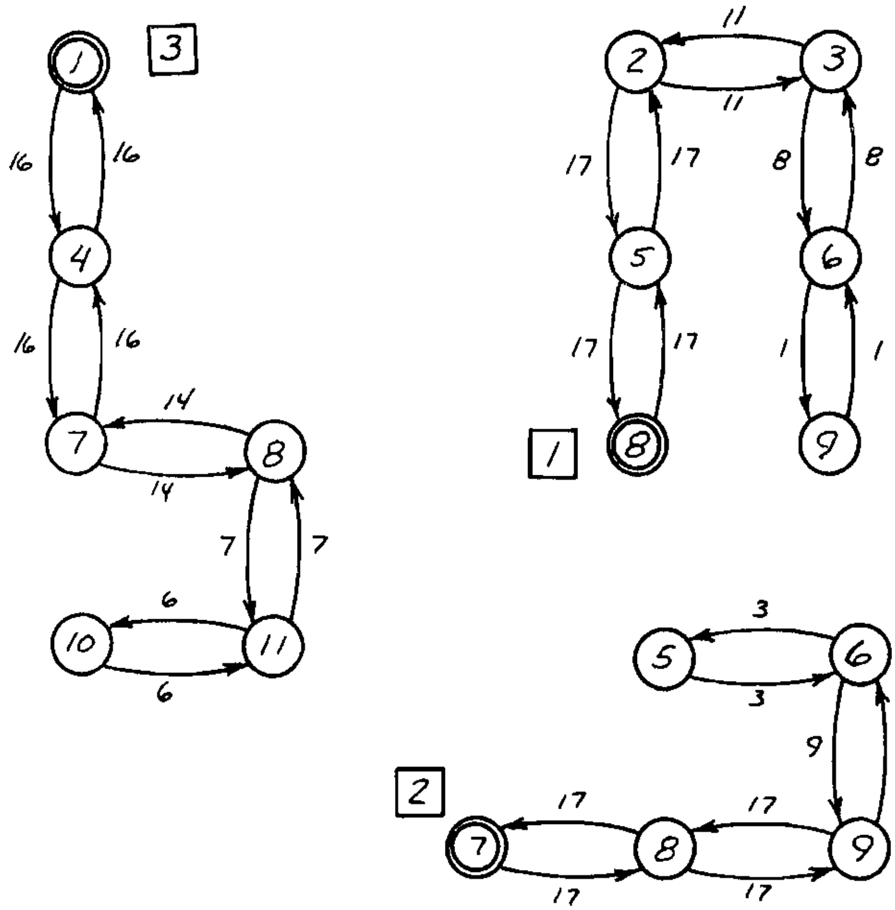
Problem D was run with three different sets of guideways, including set D1 previously described in this chapter. Problem D2 involved no guideway changes, no appending, one route combining, and three route reversings, a pattern which is very similar to Sharp's except for the route restructuring. These revisions made resulted in a cost increase of $171, which is primarily due to the vehicle service structure allowed on the routes.

Likewise, problem D3's final solution is very similar to the starting set, with no guideway changes made, no appending, and no route restructuring. This solution is approximately the same as Sharp's with a cost decrease of $48, which shows the problems involved in restructuring routes. Problem D2 had some positive route revisions but ultimately resulted in a cost increase while problem D3 had no route restructuring revisions and achieved a cost decrease. A need is shown for more accurate arc insertion parameters since D2 and D3 are very similar with D2 having three more open guideways and a final cost decrease of $1069 over D3. A need is also shown for more accurate arc deletion parameters by comparing the solutions for D1 and D2, D2 having a cost reduction of $1943 over



Travel Time Costs	= $13830
Vehicle Operating Costs	= $ 6402
Guideway Fixed Costs	= $ 1900
Sum of Fixed Plus Variable Costs	= $22132

Figure 46. Test Problem C1, Starting with C1a -- Final Routes, Vehicle Service Frequencies, and Costs



Travel Time Costs	= $14127
Vehicle Operating Costs	= $ 6776
Guideway Fixed Costs	= $ 1760
Sum of Fixed Plus Variable Costs	= $22663

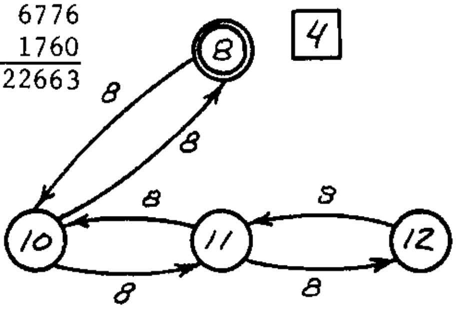
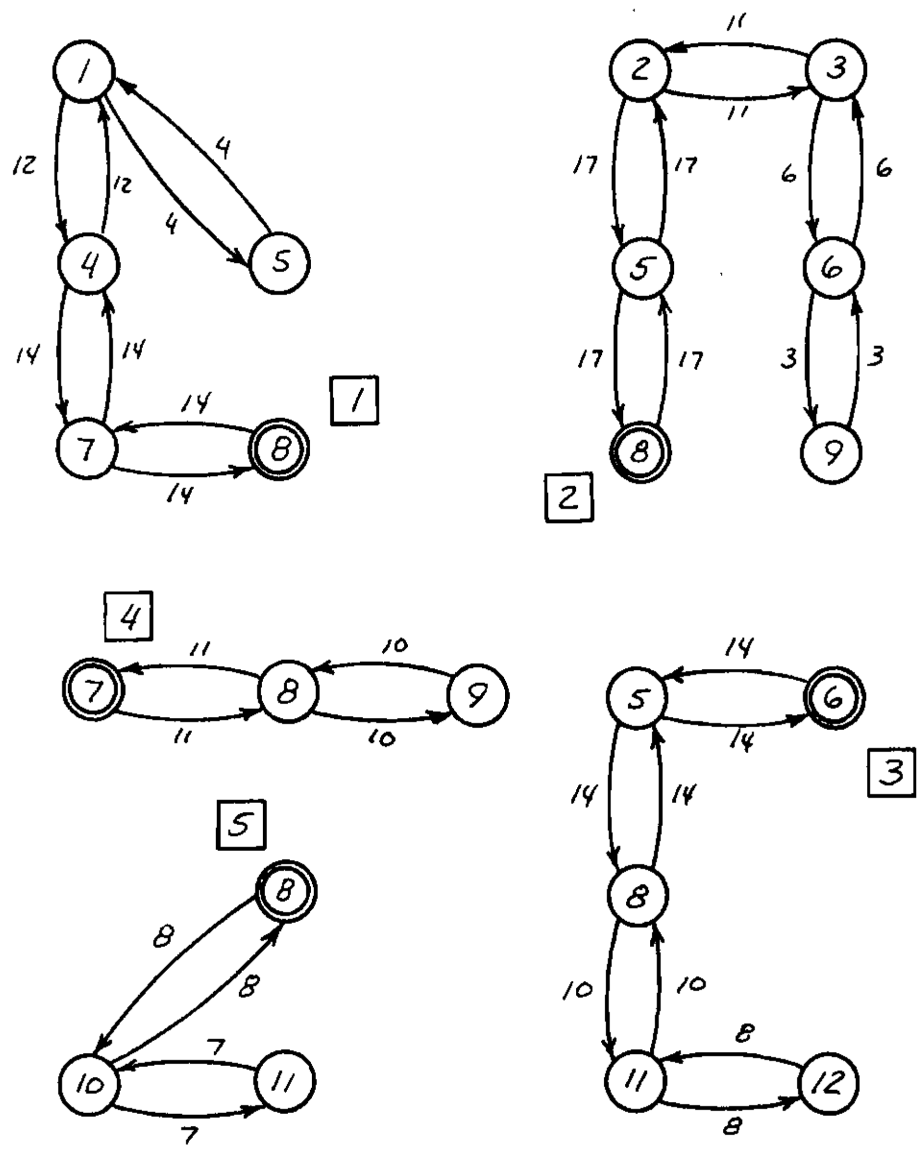
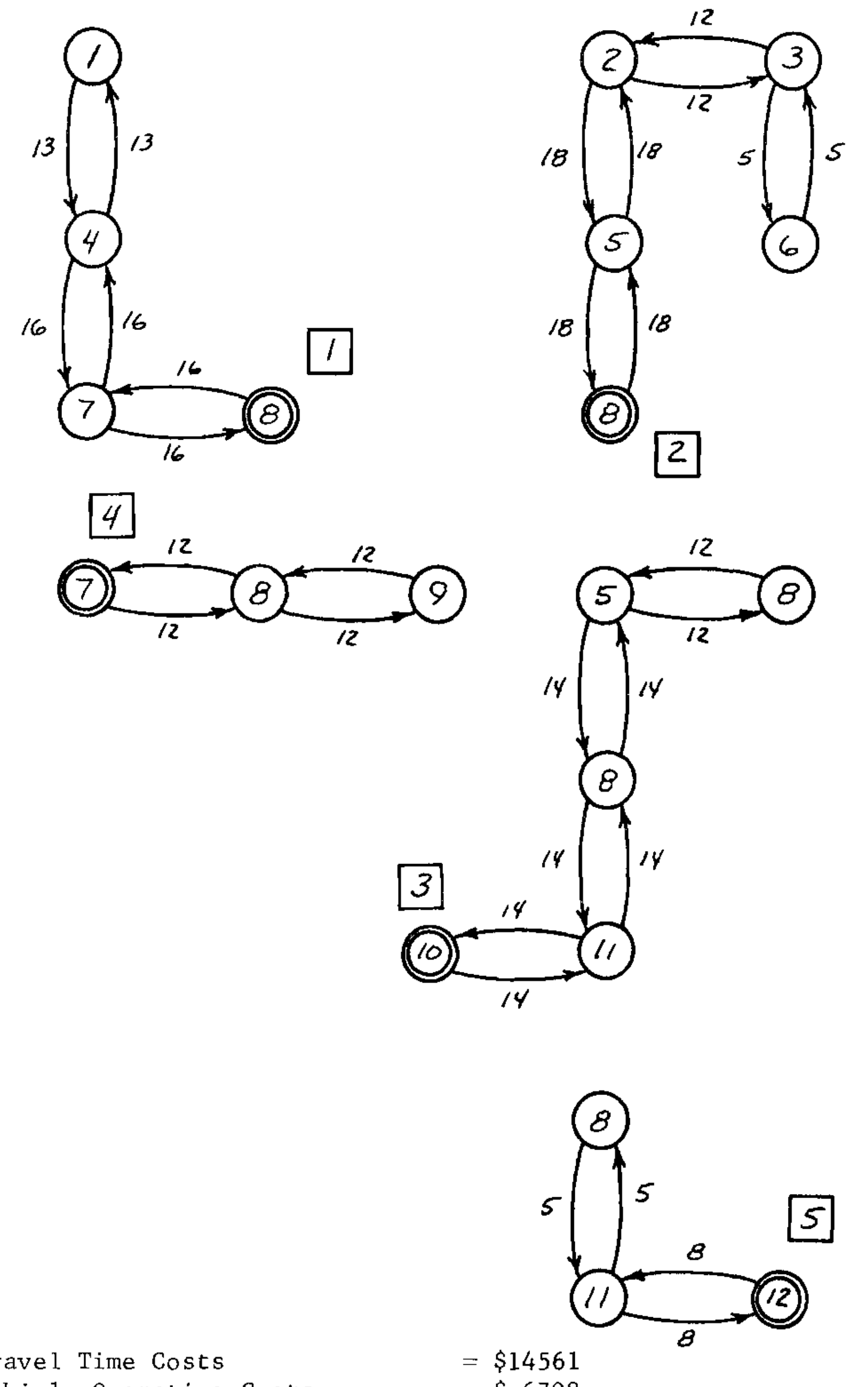


Figure 47. Test Problem C1, Starting with C1c -- Final Routes, Vehicle Service Frequencies, and Costs



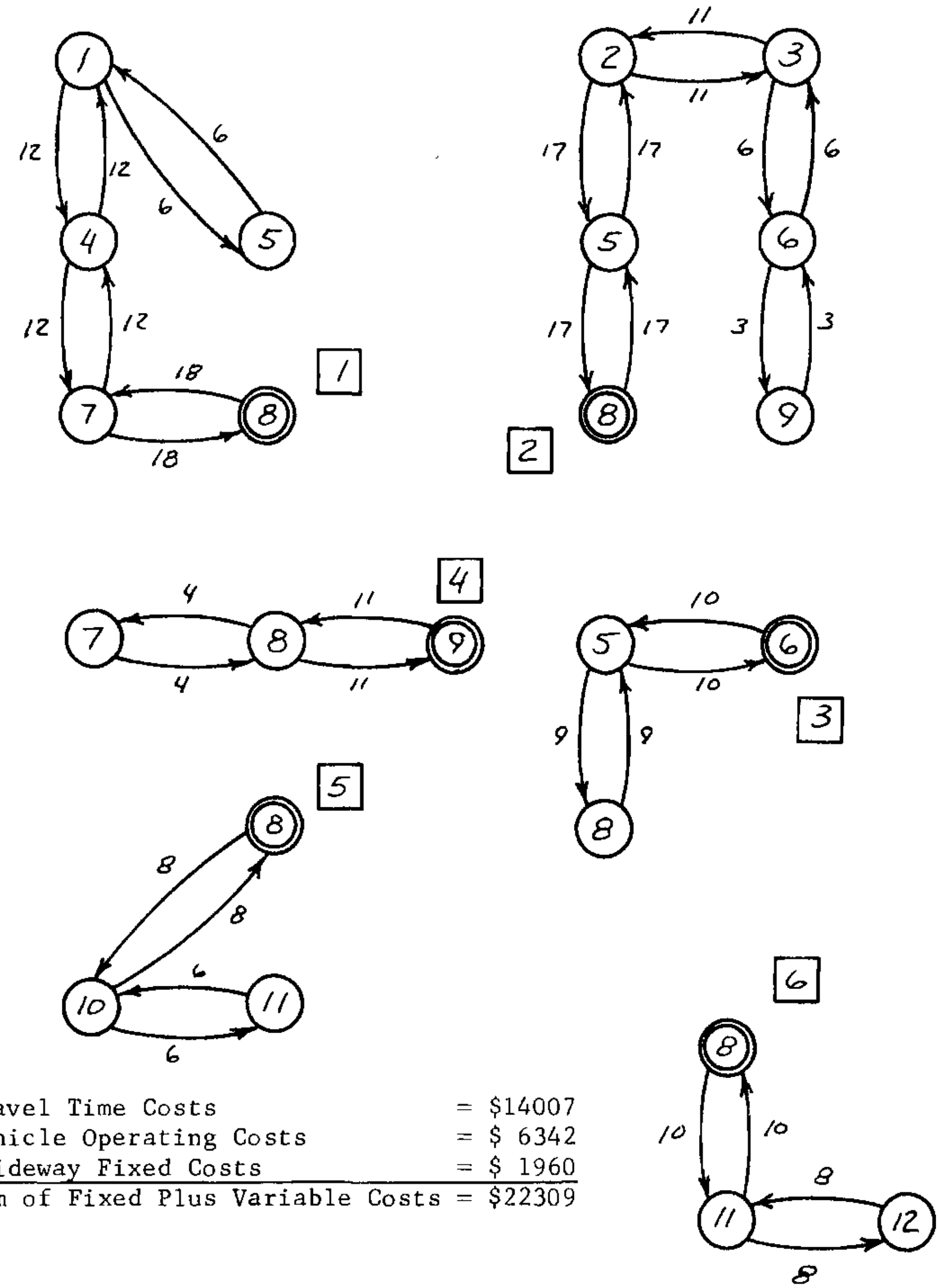
Travel Time Costs	= \$13801
Vehicle Operating Costs	= \$ 6700
Guideway Fixed Costs	= \$ 1960
Sum of Fixed Plus Variable Costs	= \$22461

Figure 48. Test Problem C2 -- Final Set of Routes, Vehicle Service Frequencies, and Costs



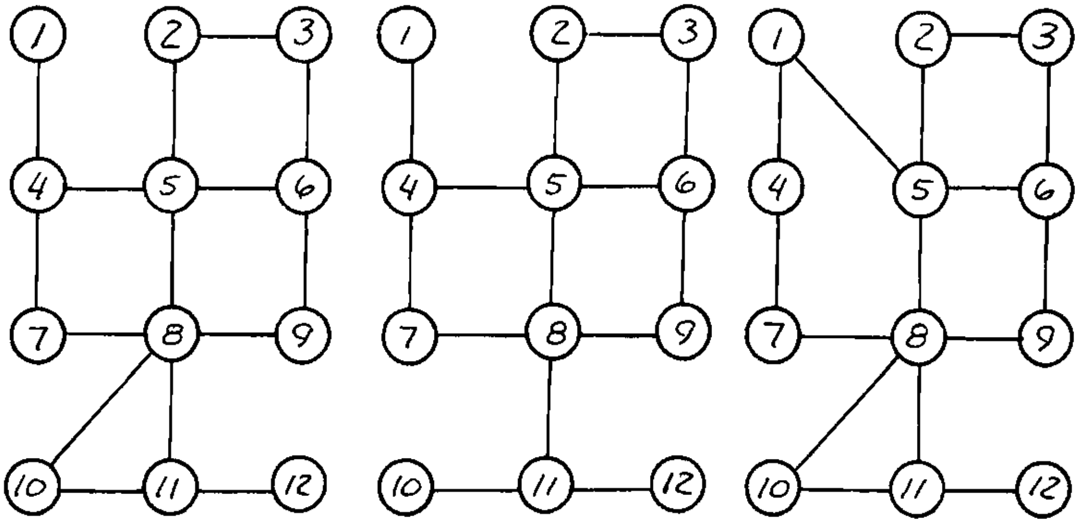
Travel Time Costs	= $14561
Vehicle Operating Costs	= $ 6708
Guideway Fixed Costs	= $ 1430
Sum of Fixed Plus Variable Costs	= $22699

Figure 49. Test Problem C3 -- Final Set of Routes, Vehicle Service Frequencies, and Costs

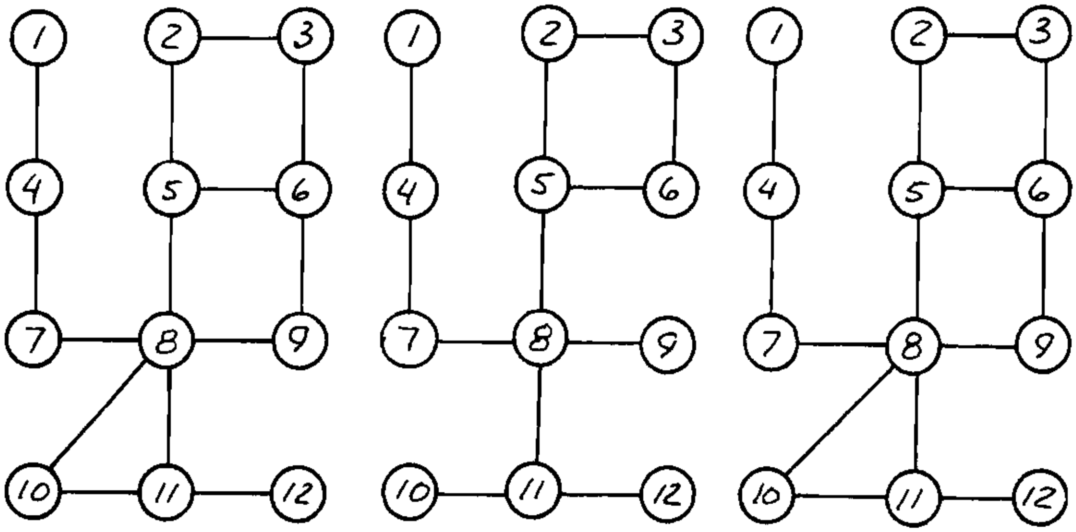


Travel Time Costs	= $14007
Vehicle Operating Costs	= $ 6342
Guideway Fixed Costs	= $ 1960
Sum of Fixed Plus Variable Costs	= $22309

Figure 50. Test Problem C4 -- Final Set of Routes, Vehicle Service Frequencies, and Costs



C1a $22,530 C1b $23,591 C1c $22,909



C2 $22,623 C3 $22,952 C4 $22,618

Figure 51. Test Problem C -- Final Sets of Open Guideways and Total Costs for Sharp's Program

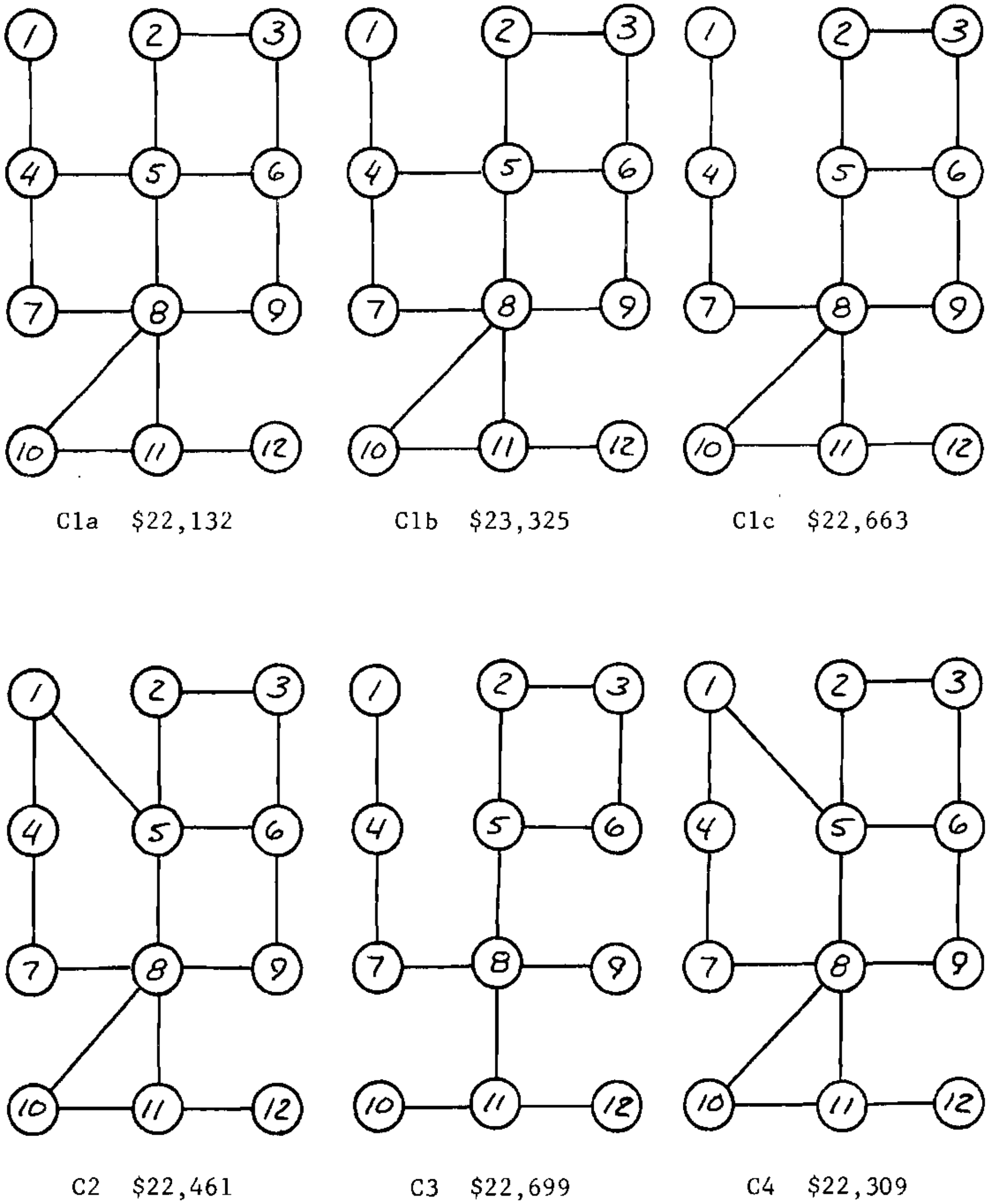


Figure 52. Test Problem C -- Final Sets of Open Guideways and Total Costs, Various Runs

D1 mainly due to deleted guideways. The initial open guideways for D2 and D3 are shown in Figure 53, initial routes and service frequencies in Figures 54 and 55, and the final routes, vehicle service frequencies, and costs in Figures 56 and 57. Sharp's final open guideways and costs are shown in Figure 58.

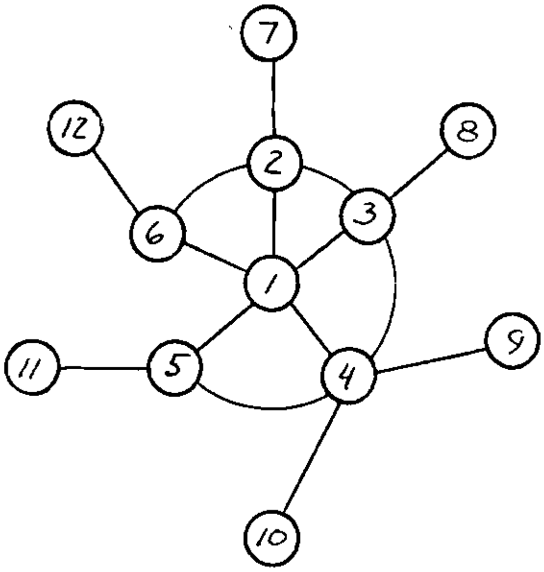
Summary of Computational Results

The results of all the problems run for this program and Sharp's program are shown in Tables 5 and 6. The items shown are number of final routes, passenger, vehicle, and guideway costs, and number of guideway algorithm iterations.

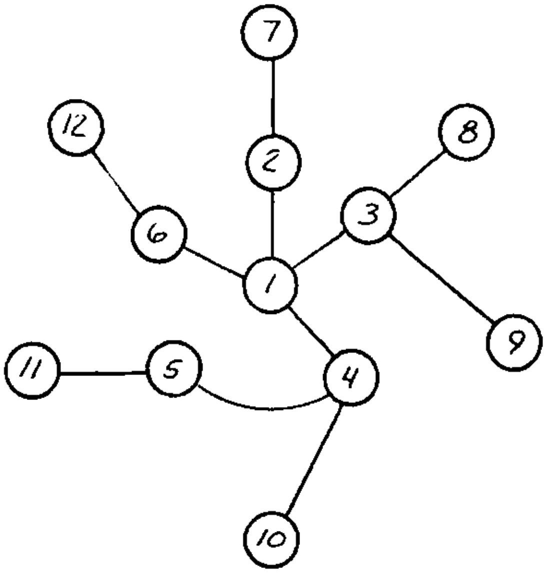
As can be seen in the comparison, the route restructuring algorithm generally resulted in lower costs (average 3% reduction), fewer final routes, and fewer guideway iterations. However, the computation times were generally higher (average 3% increase) even with the decrease in guideway iterations. This results from the multicommodity flow assignment algorithm being used more often to assign flow on proposed routes. The lower costs were primarily due to reduced passenger travel time costs, resulting from more direct service and fewer transfers.

It is interesting to note that in no case did the route restructuring algorithm lead to lower guideway costs. Vehicle operating costs were also usually higher. Thus, the overall effect of the route restructuring algorithm in reducing overall costs follows from lowering passenger time costs.

The above results of final networks and routes give a comparison of the two route algorithms within the same guideway algorithm. A more



D2



D3

Figure 53. Test Problem D -- Initial Sets of Open Guideways

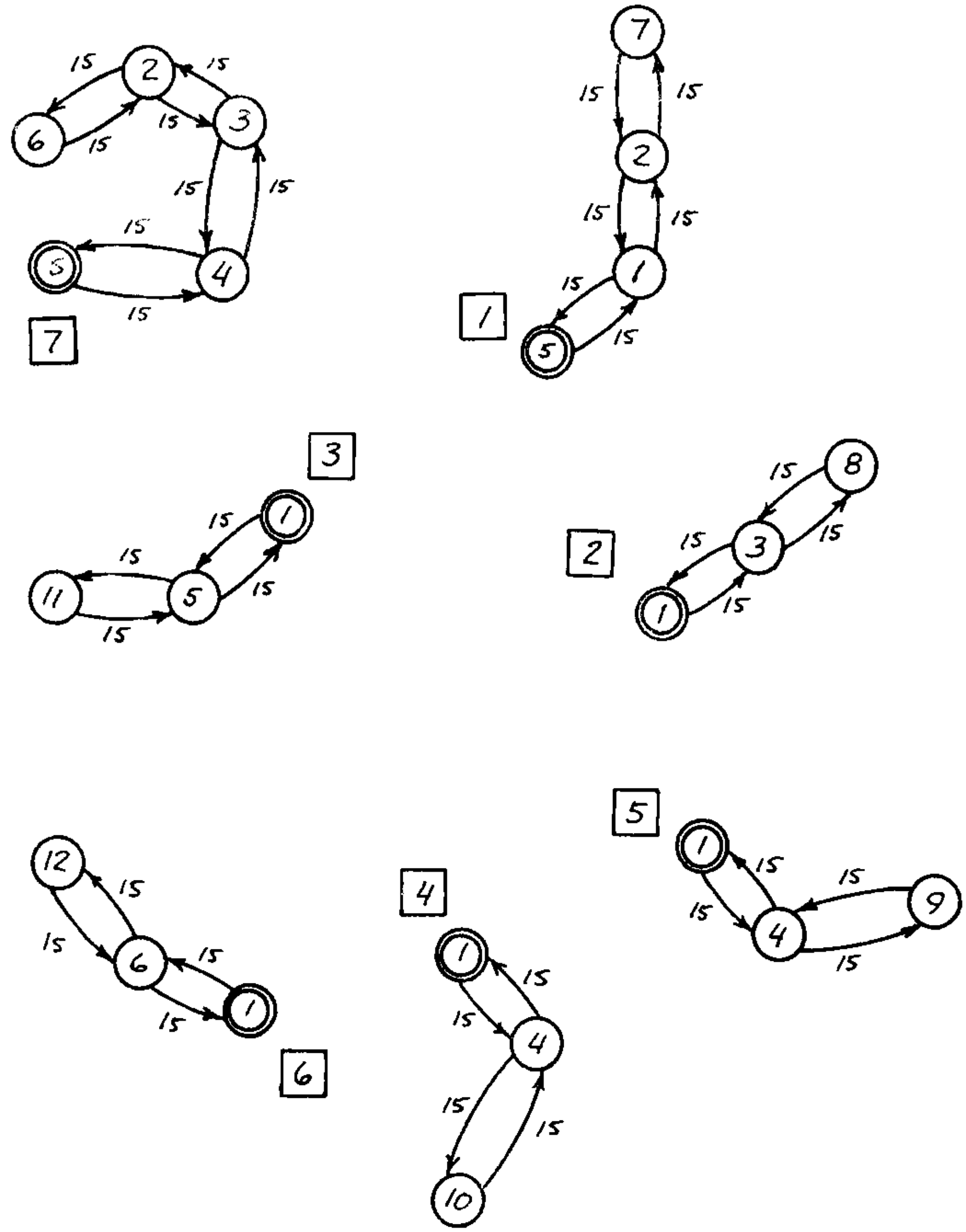


Figure 54. Test Problem D2 -- Initial Set of Routes and Vehicle Service Frequencies

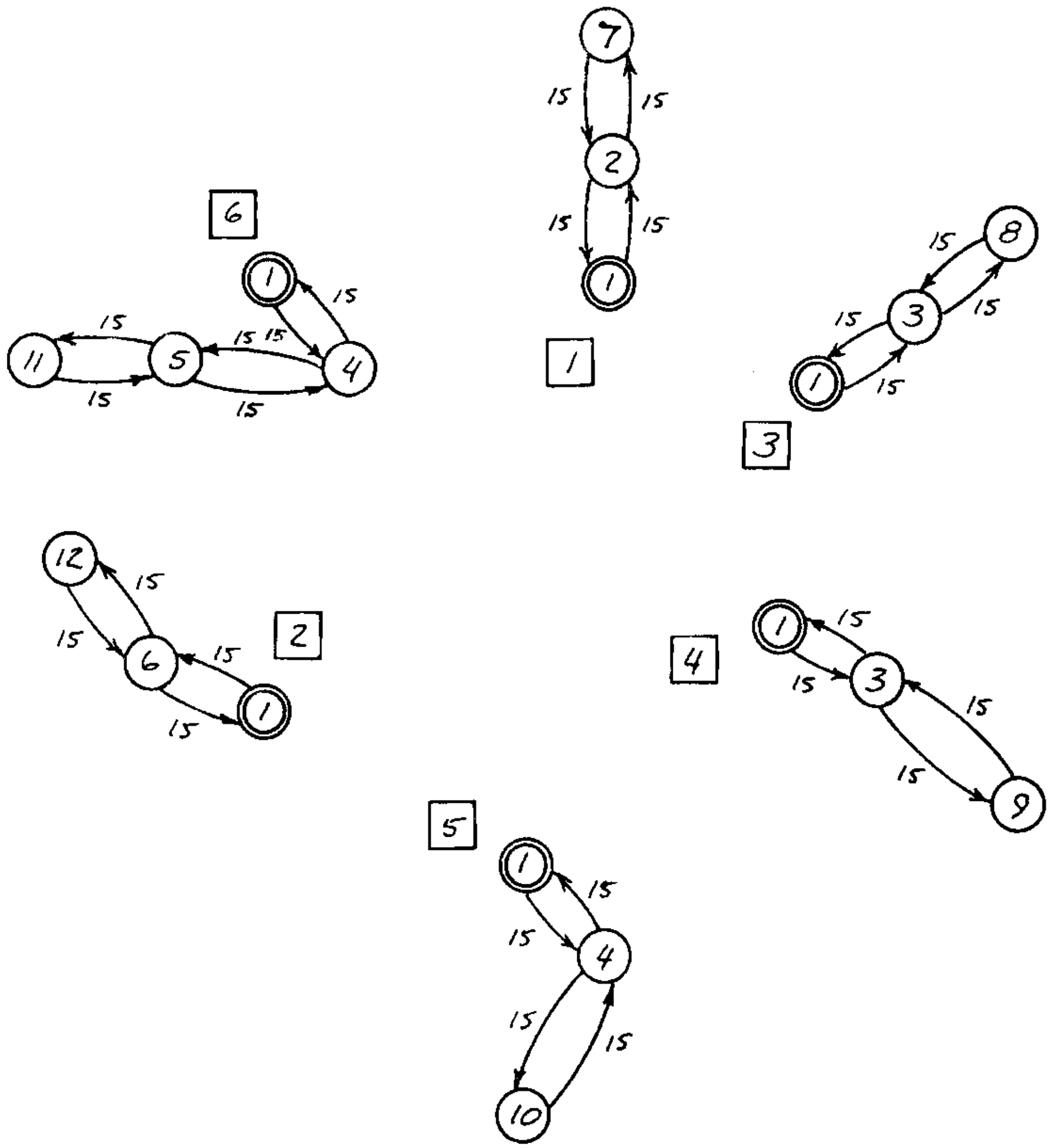
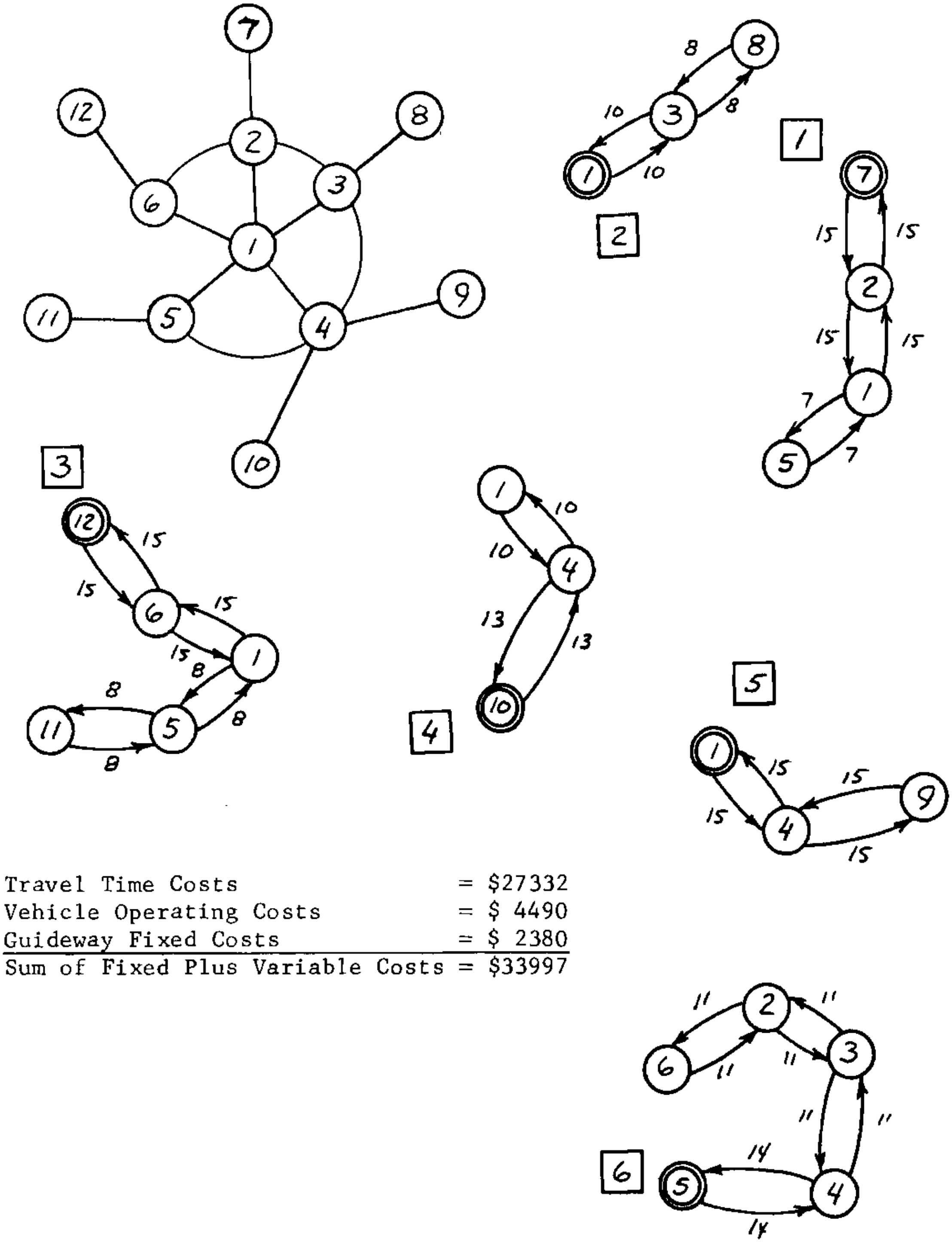
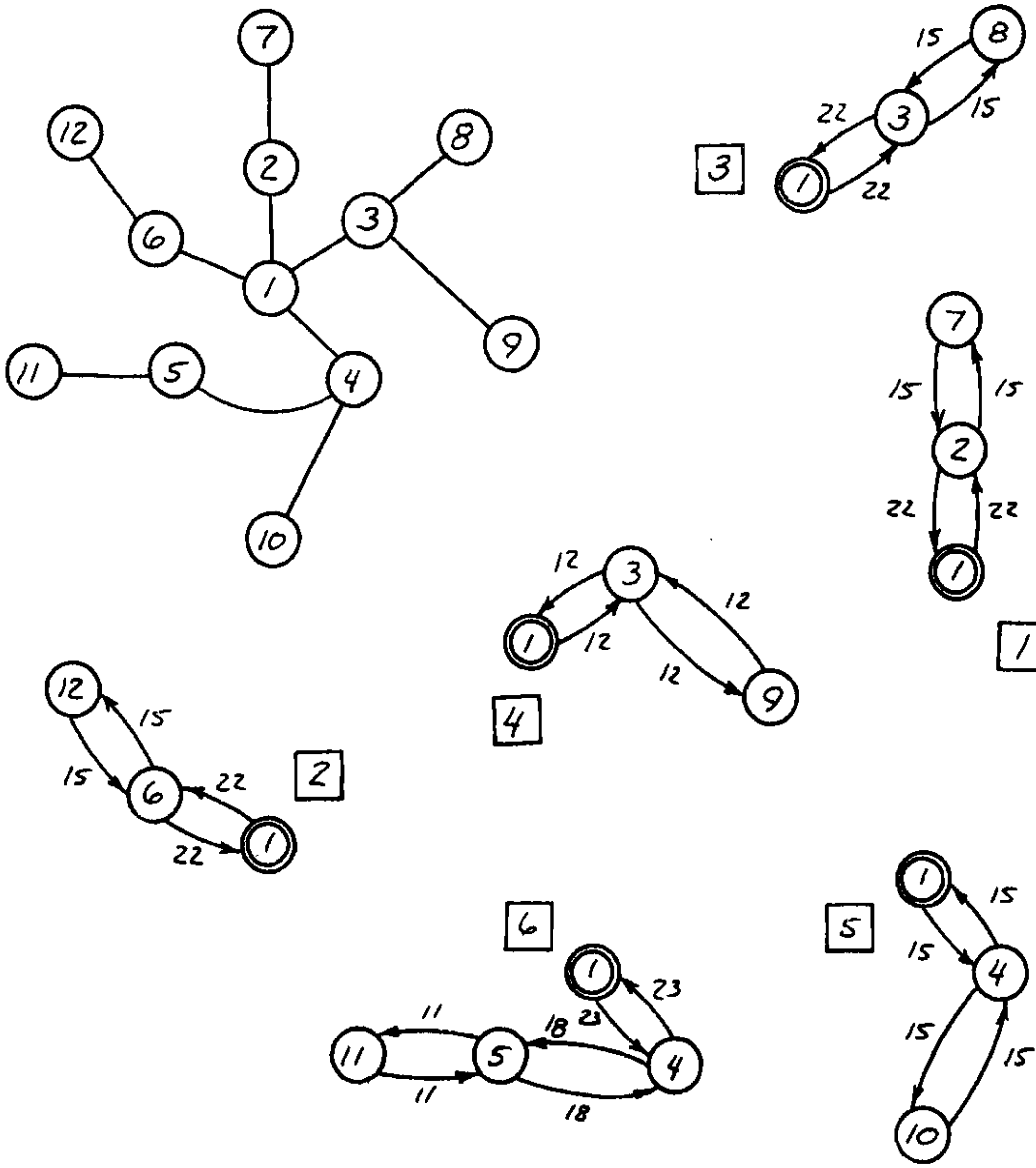


Figure 55. Test Problem D3 -- Initial Set of Routes and Vehicle Service Frequencies



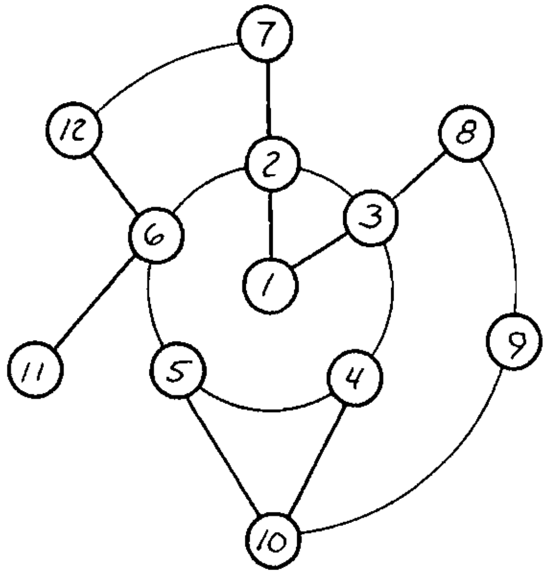
Travel Time Costs	= $27332
Vehicle Operating Costs	= $ 4490
Guideway Fixed Costs	= $ 2380
Sum of Fixed Plus Variable Costs	= $33997

Figure 56. Test Problem D2 -- Final Routes, Vehicle Service Frequencies, and Costs

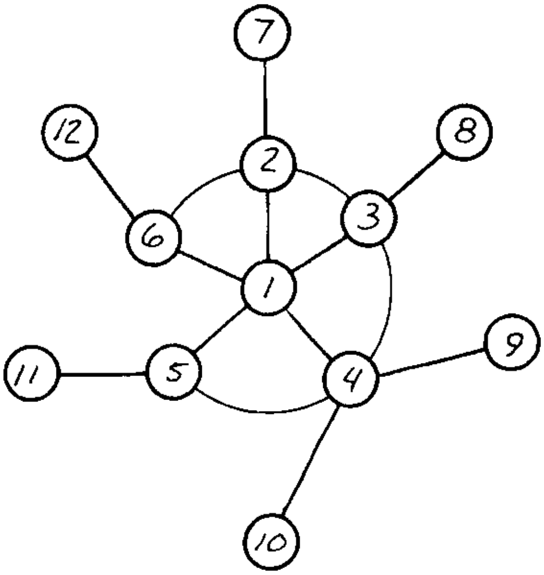


Travel Time Costs	= $28784
Vehicle Operating Costs	= $ 4592
Guideway Fixed Costs	= $ 1690
Sum of Fixed Plus Variable Costs	= $35066

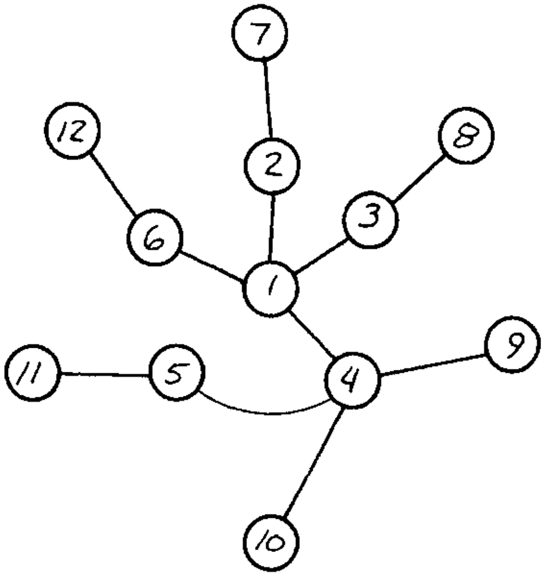
Figure 57. Test Problem D3 -- Final Routes, Vehicle Service Frequencies, and Costs



D1a $37,107



D2 $33,826



D3 $35,114

Figure 58. Test Problem D -- Final Sets of Guideways and Total Costs from Sharp's Program

Table 5. Final Results of Sharp's Algorithm -- Various Runs

Problem	No. of Initial Routes	No. of Final Routes	Passenger Time Costs	Vehicle Operating Costs	Guideway Fixed Costs	Total	No. of Guideway Iterations	Time (min)
C1a	6	7	14226	6404	1900	22530	3	.97
C1b	12	11	15659	6282	1650	23591	5	2.48
C1c	1	5	14437	6512	1960	22909	7	7.36
C2	6	6	14435	6428	1760	22673	9	2.22
C3	6	6	14858	6664	1430	22952	6	2.20
C4	6	6	14470	6388	1760	22618	5	1.38
D1a	5	6	28657	5870	2820	37107	10	18.12
D2	7	7	26606	4840	2380	33828	2	1.07
D3	6	6	28720	4704	1690	35114	2	4.02
Average	6	7	19119	6005	1928	27052	5	4.42

Table 6. Final Results of Solution Procedure -- Various Runs

Problem	No. of Initial Routes	No. of Final Routes	Passenger Time Costs	Vehicle Operating Costs	Guideway Fixed Costs	Total	No. of Guideway Iterations	Time (min)
C1a	6	6	13830	6402	1900	22132	3	1.1
C1b	12	7	14885	6540	1900	23325	3	2.3
C1c	1	4	14127	6776	1760	22663	5	8.0
C2	6	5	13801	6700	1960	22461	12	5.3
C3	6	5	14561	6708	1430	22699	6	2.5
C4	6	6	14007	6342	1960	22309	3	2.1
D1a	5	4	26742	5708	3460	35910	4	11.4
D2	7	6	22732	4490	2380	33997	2	3.5
D3	6	6	28784	4592	1690	35000	2	5.0
Average	6	5	18163	6029	2049	26241	4	4.57

direct evaluation involves examining the route structures that are obtained at one stage of the guideway algorithm. In particular, Table 7 shows such results for the zeroth guideway iteration for the same problem as listed in Tables 5 and 6.

This comparison of Sharp's route algorithm with the route restructuring algorithm shows varied results. Computation time is generally less for Sharp's program, again due to the multicommodity flow assignment algorithm. The route restructuring algorithm generally achieved lower total costs, with the savings from test problems E and C in passenger travel time costs, and for problem D the savings in vehicle operating costs. The average reduction in travel time and vehicle operating costs was 1% here.

Table 7. Comparison of Sharp's Route Algorithm to Route Restructuring Algorithm at End of Guideway Iteration Number Zero

Problem	Passenger Time Costs		Vehicle Operating Costs		Sum of Passenger and Vehicle Costs	
	Sharp's	Bell's	Sharp's	Bell's	Sharp's	Bell's
E6	17735	17721	5434	5134	23169	22855
Cla	14426	13830	6404	6402	20830	20232
Clb	15616	14885	6082	6540	21968	21425
Clc	14296	14296	7464	7464	21760	21760
C2	14435	14031	6428	6692	20863	20723
C3	16416	16506	7218	6966	23634	23472
C4	14193	14007	6496	6342	20689	20349
Dla	27346	29100	7597	5384	34943	34484
D2	26606	27332	4840	4490	31446	31822
D3	28720	28784	4704	4592	33424	33376

Table 7. (Continued)

Problem	Number of Route Algorithm Iterations		Final Number		Computation Time (seconds)	
	Sharp's	Bell's	Sharp's	Bell's	Sharp's	Bell's
E6	8	6	4	4	6	5
C1a	21	16	7	6		27
C1b	24	39	12	7	38	92
C1c	35	33	5	5	326	374
C2	27	24	6+4(0)	5+5(0)		70
C3	9	9	6	6		9
C4	17	22	6	6		43
D1a	26	16	7	4+2(0)	263	459
D2	8	3	7	6	7	43
D3	24	25	6	6	116	136

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The area of public transit planning optimization is important, both in terms of public service and monetary rewards. Any small improvement achieved in any aspect of this system can account for large cost reduction.

This research focused on extending Sharp's work in public transit network design in two main areas, route restructuring and multiple guideway changes. Sharp's computer program has been modified and extended to incorporate these changes. The algorithm still remains heuristic, as no attempt was made to obtain an optimal solution for the large problems typically encountered.

The results obtained from this program generally achieved a reduction in the final number of routes and a reduction of 3% in final total costs with an increase of 3% in computational time. In a majority of cases the cost decrease resulted from lower passenger travel time costs.

The multiple guideway algorithm obtained a reduction in final costs in some test problems and no change occurred in the others. For the problems which remained the same, the initial guideway system was the best solution as no guideways were inserted or deleted.

Recommendations

Based on the computational experience with this improved algorithm, the following areas have been identified as areas for future research:

1) Improved criteria for the selection of guideways to be inserted and deleted need to be developed. More accurate guideway improvement parameters would reduce the computation time and generally obtain better results.

2) An interactive program needs to be written to enable the user to insert, delete, or modify routes as guideways are inserted and deleted. Many intuitive changes can be made by the user that would be cumbersome to program and time consuming for the computer. Also, more use could be made of guideways when they are inserted, and to restructure routes as guideways are deleted.

3) A program needs to be incorporated to generate a good initial set of routes based on a function of shortest travel and delay time costs. This would give the user an idea of where the main flow of passengers would like to travel, which will aid during the interactive phase of restructuring routes.

4) Route structures should be modified to allow major stations to occur anywhere in the route and not just at the beginning. Vehicle service frequency would then be greatest at the arcs from the major station and weakly decreasing towards the ends. This would provide a better matching of vehicle service with CBD-oriented demand, either by eliminating excess service on route ends or by reducing transfers on cross-town trips.

BIBLIOGRAPHY

1. Aburto-Avila, Jose Luis, Optimal Design of Transportation Networks with Fluctuating Demand -- A Case in Multicommodity Network Flows, Ph.D. Thesis, Stanford University, 1973.

2. Bartlett, T. E., and A. Charnes, "Cyclic Scheduling and Combinatorial Topology; Assignment and Routing of Motive Power to Meet Scheduling and Maintenance Requirements; Part II, Generalization and Analysis," Naval Research Logistics Quarterly, Vol. 4, No. 3, 207-220, 1957.

3. Bergendahl, Göran, "A Combined Linear and Dynamic Programming Model for Interdependent Road Investment Planning," Transportation Research, Vol. 3, No. 2, 211-228, 1969.

4. Billheimer, John W., Optimal Route Configurations with Fixed Link Construction Costs, Ph.D. Thesis, Stanford University, 1970.

5. Christofides, N., and S. Eilon, "An Algorithm for the Vehicle Dispatching Problem," Operating Research, Vol. 20, No. 3, 309-318, 1972.

6. Ford, L. R., and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

7. Gaskell, T. J., "Bases for Vehicle Fleet Scheduling," Operations Research, Vol. 18, No. 3, 281-295, 1970.

8. Hyman, Warren, and Larry Gordon, "Commercial Airline Scheduling Technique," Transportation Research, Vol. 2, No. 1, 23-29, 1968.

9. Ishmael, P. C., Synthesis of Minimum Cost or Maximum Service Rapid Transit When Travel Time Cost Is Included, Ph.D. Thesis, Arizona State University, 1972.

10. Ochoa-Rosso, Felipe, Applications of Discrete Optimization Techniques to Capital Investment and Network Synthesis Problems, Search and Choice in Transport Systems Planning, Vol. 3, Dept. of Civil Engineering, Massachusetts Institute of Technology, 1968.

11. O'Connor, Arthur D., and Charles DeWald, A Sequential Deletion Algorithm for the Design of Optimal Transportation Networks, Paper presented at the 37th National ORSA Meeting, Washington, D. C., April 1970.

BIBLIOGRAPHY (Concluded)

12. Scott, Allen J., "The Optimal Network Problem: Some Computational Procedures," Transportation Research, Vol. 3, No. 2, 201-210, 1969.

13. Sharp, G. P., Public Transit System Network Models: Consideration of Guideway Construction, Passenger Travel and Delay Time, and Vehicle Scheduling Costs, Ph.D. Thesis, Georgia Institute of Technology, 1974.

14. U. S. Department of Commerce, Traffic Assignment Manual, Bureau of Public Roads, Washington, D. C., June 1964.

15. Univac 1100 Series FORTRAN V Programmer Reference Manual, UP-4060 Rev. 2, Sperry Rand Corporation, 1972.

16. Urban Mass Transit Administration, UMTA Transportation Planning System, NTIS No. PB 212-930, September 1972.

17. Yaged, B., Jr., "Minimum Cost Routing for Static Network Models," Networks, Vol. 1, No. 2, 139-172, 1972.

18. Young, Dennis R., "Scheduling a Fixed-Schedule, Common Carrier Passenger Transportation System," Transportation Science, Vol. 4, No. 3, 243-269, 1970.