

PUBLIC SECTOR VEHICLE ROUTING:
THE CHINESE POSTMAN PROBLEM

by

ROBERT STRICKER

B.S., Tufts University
(1969)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August, 1970

Signature of Author
Department of Electrical Engineering, August 24, 1970

Certified by
Thesis Supervisor

Accepted by
Chairman, Departmental Committee on Graduate Students

Archives



PUBLIC SECTOR VEHICLE ROUTING:
THE CHINESE POSTMAN PROBLEM

by

ROBERT STRICKER

Submitted to the Department of Electrical Engineering on August 24, 1970
in partial fulfillment of the requirements for the Degree of Master of
Science.

ABSTRACT

This research is concerned with vehicle routing problems in the public sector. Specifically, the problems of routing vehicles to service every street in a complex highway network are analyzed. Chapter I presents a general overview of what these problems are, and discusses the general theoretical problem known in the literature as the Chinese postman problem. Chapter II analyzes the specific problems of trash collection and snow plowing, and demonstrates how these problems are in fact variations of the Chinese postman problem which require the consideration of additional constraints such as those imposed by limited vehicle capacity. Chapter III gives a brief review of existing algorithms for the Chinese postman problem, while Chapter IV describes some new algorithms proposed by the author, including one which is an attempt at handling the capacity constraint. Finally, Chapter V offers some computational experience on a large scale problem using one of these proposed new algorithms. Results indicate that the algorithm used in Chapter V is a feasible way to solve certain vehicle routing problems for large cities, but more research remains to be done in the search for more efficient and more versatile algorithms.

THESIS SUPERVISOR: David H. Marks
TITLE: Assistant Professor of Civil Engineering

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor, Professor David H. Marks, for his supervision of this work. His constant encouragement, helpful suggestions, and good advice played a key role during the past year while this research was being conducted.

I would also like to thank my wife Jane, whose criticism of the final draft helped to make this thesis more readable. Finally, I would like to thank my Faculty Counselor, Professor Alvin Drake, my supervisor at Bell Telephone Laboratories, Mr. Robert L. Potter, and all others who were so helpful to me during the course of my graduate education.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	2
ACKNOWLEDGEMENT	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
CHAPTER I. INTRODUCTION TO VEHICLE ROUTING PROBLEMS	7
1. General Overview	7
2. Formulation of the Chinese Postman Problem	10
CHAPTER II. ANALYSIS OF THE TRASH COLLECTING AND SNOW PLOWING PROBLEMS FOR THE CITY OF CAMBRIDGE, MASSACHUSETTS	13
1. Analysis of the Trash Collection Problem	13
2. Analysis of the Snow Plowing Problem	19
CHAPTER III. SURVEY OF EXISTING SOLUTION TECHNIQUES FOR THE CHINESE POSTMAN PROBLEM	21
1. Existence of an Euler Tour	21
2. Mei-Ko	22
3. Glover's Pseudo Edges	23
4. Murty's Symmetric Assignment Problem	25
5. Edmonds' Matching Algorithm	28
6. Johnson's Existence Theorem for an Euler Tour on a Bidirected Graph	31
7. Liebman and Marks' Travelling Salesman Approach	33
8. Glossary	35

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
CHAPTER IV. PROPOSED NEW SOLUTION TECHNIQUES	38
1. Modification of Murty's Assignment Algorithm	38
2. Decomposition Algorithm	41
3. Solution of m-Postmen Chinese Postman Problem	47
CHAPTER V. COMPUTATIONAL EXPERIENCE ON A LARGE SCALE PROBLEM	52
1. Problem Description	52
2. Computational Procedure	53
3. Results	61
CHAPTER VI. CONCLUSIONS AND RECOMMENDATIONS	69
APPENDIX I. CONSTRUCTION OF AN EULER CHAIN	72
APPENDIX II. GLOVER'S ALGORITHM	76
APPENDIX III. MURTY'S MATRIX REDUCTION METHOD AND BRANCHING RULES	84
BIBLIOGRAPHY	88

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
4-1	Example of a Symmetric Cost Matrix with Unsymmetric Optimal Assignment	39
4-2	Example of Decomposition Algorithm	45
4-3	Ad Hoc Partitioning of a Network	48
4-4	m-Postmen Example	50
4-5	Solution of m-Postmen Example	50
5-1	Map of City of Cambridge, Massachusetts	54
5-2	Map of Southwest Portion of Cambridge, Massachusetts	55
5-3	Map Showing Partitioning of the Network into Three Sections	56
5-4	Map of Section I Including Boundary Edges	57
5-5	Map of Section II Including Boundary Edges	58
5-6	Map of Section III Including Boundary Edges	59
5-7	Solution of the Chinese Postman Problem for Section I	62
5-8	Solution of the Chinese Postman Problem for Section II	63
5-9	Solution of the Chinese Postman Problem for Section III	64
5-10	Solution of the Chinese Postman Problem for Entire Network	65
5-11	Map of Area Used as Basis for Comparison	67
I-1	Example of a Graph with Two Odd Nodes	74
I-2	Euler Tour for Network in Figure I-1	74

CHAPTER I. INTRODUCTION TO VEHICLE ROUTING PROBLEMS

1. General Overview

The problems faced by local, state, and federal governments in meeting their citizens' needs such as education, transportation, and a clean environment, are known as public sector problems. Problems faced by industry in their quest for more efficient operating procedures and greater profits are known as private sector problems. Until recently, most of the effort in the field of operations research has been concentrated on private sector problems. There are two major reasons for this. The first reason is that the objective in the private sector is generally much easier to express. It almost exclusively is to maximize profits or to minimize costs, where benefits and costs are clearly defined and quantifiable in commensurate units. In the public sector, on the other hand, people often cannot even agree on objectives, let alone quantify them. This is because intangibles of a political and social nature are involved. Currently, research is being conducted to define public sector problems more precisely.

The second reason for the lack of research in public sector problems is due to the inherent size of these problems. A private contractor who desires to optimize his operation usually must service relatively few customers, and thus faces what is basically a discrete problem, since each individual demand can be handled separately. Government, on the other hand, has a different objective, that is to serve the public in

general, so it has much greater demands to be taken into consideration, and the optimal operation problem turns into a continuous one because there are too many demands to treat each one individually. Generally, the discrete problem is easier to solve. Fortunately, though, with the development of more sophisticated operations research techniques and the availability of modern high speed computers, it now becomes possible to attempt to solve these large scale problems, both continuous as well as discrete. This thesis will be concerned primarily with the public sector.

One class of problems encountered in the analysis of the public sector involves the routing of vehicles. This problem is basic to such city services as trash collection, street cleaning, postal delivery, and meter reading for municipal power companies. The basic problem involved in all these services is finding optimal least cost routes for vehicles which must service every street in a complex road network. The related general theoretical problem, known in graph theory as the Chinese postman problem, is to trace the shortest continuous path through a network so that every arc is covered at least once. Edmonds^[8], Glover^[10], and Murty^[21] offer algorithms with three different approaches to solve the theoretical problem.

Unfortunately, the Chinese postman problem is not an adequate model for many important city services. In trash collection, for instance, each vehicle is subject to a capacity constraint so that a single truck cannot possibly service an entire city in one trip. A truck must return to the city dump or incinerator each time it is full before it can resume

its tour. This means finding many distinct continuous tours, each with a common origin, rather than just one tour. In snow plowing, time constraints cause the use of more than one vehicle with the added constraint requiring all primary roads to be plowed before any secondary roads, which in turn must all be plowed before any residential streets are plowed. The Chinese postman problem of finding a single shortest continuous path covering every arc in the network is obviously not capable of handling these added constraints.

Certain additional constraints must also be given consideration in defining these public sector problems. In trash collection, how often will trash be collected, how many men and trucks will be needed, should transfer sites be used, and what type of disposal facility is best? In snow plowing, when should the plowing start, when should it end, and how many men are needed? All these questions must be answered, whether by some form of ad hoc reasoning as is presently the case, or by using analytical techniques such as those for trash collection suggested by Marks and Liebman^[17]. The cost of collecting trash from a given street, for instance, will depend on the frequency of collection because the more often the street is serviced, the less trash will accumulate, so the faster the truck can move along. With a lighter load, perhaps even fewer men will be required for a crew. There is also the sociological effects of increased collection frequency to be considered. But do these advantages compensate for the increased costs of more frequent collections? This thesis will ignore these very important questions, and instead will be concerned with finding the optimal routing subject to a given policy.

Certainly the solution techniques developed in this thesis may be a useful tool in the analysis for determining the optimal policy to use. By varying policies and finding optimal routes in each case, costs can be determined for each policy, and the optimal policy selected.

2. Formulation of the Chinese Postman Problem

The recurrent theme in these routing problems is to minimize the total distance travelled in describing an edge covering tour, i.e., in traversing every edge in the network at least once. Although the Chinese postman problem alone is not an adequate model for most real world problems, by studying solution techniques for the theoretical problem, adaptations of these techniques may become apparent which lend themselves to the real world problems. With this in mind, the following general formulation of the Chinese postman problem is presented since it is the framework the practical problems will build from.

The Chinese Postman Problem

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N C_{ij} X_{ij} \quad (1)$$

subject to:

$$\sum_{k=1}^N X_{ki} - \sum_{k=1}^N X_{ik} = 0 \quad i=1, \dots, N \quad (2)$$

$$X_{ij} + X_{ji} \geq 1 \quad \text{for all arcs } (i,j) \in A \quad (3)$$

$$X_{ij} \geq 0 \quad \text{and is integer} \quad (4)$$

where

N = the number of nodes in the network

A = the set of all arcs in the network

X_{ij} = the number of times the arc from node i to node j is traversed

C_{ij} = the length of the arc from node i to node j .

Expression (1) is the objective function which is the minimization of the distance required to travel through the network and cover every edge at least once. Equation (2) expresses the continuity of flow requirement for the network; the number of edges travelled going into any node must equal the number going out. Equation (3) states the requirement that each edge must be travelled at least once, and equation (4) is the nonnegativity restriction.

Although this formulation aids in the formal definition of the underlying problem, it does not lend itself to any efficient solution techniques. Traditional integer programming techniques such as branch and bound algorithms and simplex based cutting plane algorithms are not appropriate due to the size of these problems. Since any real city has hundreds of streets, and the number of constraints as well as the number of variables is at least twice the number of streets, these algorithms with the present state of the art could not possibly handle the problem. Also, the formulation, although highly structured, does not fit into any standard network algorithms. The only existing algorithms for the problem have been developed specifically for it, or involve manipulating the problem to utilize a general solution technique. A discussion of existing algorithms will be delayed until Chapter III, so that first a

thorough analysis of some of the real problems as they actually exist can be presented. Chapter IV will present proposed new solution techniques, followed by the solution of a moderate sized sample problem in Chapter V.

CHAPTER II. ANALYSIS OF THE TRASH COLLECTION AND SNOW PLOWING PROBLEMS FOR THE CITY OF CAMBRIDGE, MASSACHUSETTS

Before one can attempt the solution of complex real world problems, it is necessary to understand what the problems and all their ramifications are. In order to gain greater insight into some of these problems, this author spent a week working with the Public Works Department for the city of Cambridge, Massachusetts. The city of Cambridge was selected due to its proximity to the author, and due to the fact that it is typical of a moderate sized city with many complex urban problems.

1. Analysis of the Trash Collection Problem

The problem of solid waste disposal is rapidly growing out of control in urban America. Although a great deal of research is being conducted into efficient means of disposal such as incineration and railroad haul, currently collection costs rather than disposal costs make up most of the budget. A study by Ludwig and Black^[18] reveals that 85 percent of the solid waste system cost in this country is due to collection while only 15 percent is due to disposal. Thus, in the short run at least, some of the financial pressure may be taken off the solid waste disposal system by improving collection efficiency. One scheme to accomplish this, which is analyzed by Marks and Liebman^[17], involves setting up local transfer facilities where the collection vehicle transfers its load to a vehicle which is more efficient for making the longer run to the disposal facility.

Certainly schemes such as this may prove beneficial for large cities at least. However, one problem which most cities seem to have ignored is efficient routing of the collection vehicles. As vehicles have become larger and more efficient, each one can service a greater area with no increase in the size of its crew. Due to the complexity of the problem, though, as collection fleets become modernized, instead of completely restructuring routes in light of increased efficiencies, city administrators merely append bits and pieces of a phased out route to remaining routes. This often results in obvious inefficiencies such as routes which are no longer contiguous. This author's investigation of the routing of collection vehicles for the city of Cambridge revealed many such inefficiencies in the layout of routes. The city is divided into five main sections, one of which is covered each weekday. Each section is further divided into fifteen subsections, each of which is intended to be covered by a separate truck. These subsections do not cover a compact area, but often have streets interspersed amongst each other, and some even cover two widely separated areas.

Moreover, during the summer of 1970 when this investigation was made, rarely were more than nine or ten city trucks in operating condition on any given day. This necessitated the expensive practice of hiring two or three private contractors, plus splitting up the remaining routes among the trucks on an overtime basis. Since it was not known until the start of the day how many trucks would be available, a superintendent would split the unmanned routes into overtime assignments as best he could off the top of his head, and then he would get in his car so he could

find each crew and hand them their assignment. Since the overtime routes were determined without even the aid of a map, and were scribbled down on scraps of paper, streets were sometimes missed which would have to be picked up by a special crew later that night. Also, although the area covered by each subsection is well defined, in what order the streets are to be covered is left to the drivers, who supposedly are familiar with the area. Because no set routes exist, a city worker must drive along in front of each contractor's crew to show them where to go. Also, the same crew generally covers the same area each week since they are familiar with the route. This causes complaints among the men since certain routes are thought to be harder than others. If only set route assignments were drawn up, these problems could be eliminated. It would be a simple matter to partition a set route assignment among the available crews before they leave in the morning. Also, it is much less expensive to hand a contractor's crew a sheet of paper with a route on it than to pay a city worker to show them the route. Finally, routes may be systematically rotated among the crews each week so that one crew does not continually get one of the less desirable routes.

It should be mentioned at this point that one innovation which has proven successful for the city of Cambridge is the use of an incentive system. Each crew, instead of working set hours, is given an area to cover. When they complete their assignment, they are free to leave. The assignments are based upon the average collection load on each street and average collection rates. Overtime pay is based on the average time which would be required to cover streets not in the original assignment. Of

course, the critical point in an incentive system is determining assignments which are fair for all parties involved. The average crew for the city of Cambridge completes its assignment between 12:30 and 1:00 instead of the normal quitting time of 4:00. However, it must be kept in mind that the men take only a short 10 or 15 minute coffee break in midmorning, and take no lunch hour. This author's experience has shown that the men seem to push themselves extremely hard in order to finish early. They no longer waste time with such common practices of the past as junking, i.e., looking through barrels for salvageable items. Based on Cambridge's experience, it would seem that the incentive system is a successful way to motivate men in a generally undesirable occupation such as trash collection.

Whether or not the incentive system is used, one still wants the most efficient way of routing vehicles through the area to be collected from. Assuming such issues as collection frequency and number of crews have been decided, what one would like to determine is the exact route for each vehicle, including going to the dump, so as to minimize the distance travelled by the entire fleet. Unfortunately, such rigid routing is inappropriate in this situation due to the great variation in distance a truck can travel before being filled. Examination of actual situations in Cambridge reveals that a truck may make two trips to the dump from a route one week, but may have to make three trips from that same route the next week. This apparent stochastic nature of refuse output is hard to explain since one would expect the trash generated should average out among the many households serviced along any particular

route. The best alternative in a situation such as this is in some manner, usually based on past experience, determine how large an area a single vehicle can service in a day, and then divide the city up into the proper sized sections. The boundaries of the sections and the routing of the vehicle within each section are chosen so as to minimize the distance travelled by the fleet. Note that while in reality each section should be further subdivided according to the number of trips the vehicle must make to the disposal facility, this must be ignored in the analysis due to the variance of this figure.

It can be seen then that the problem is somewhat more involved than the theoretical Chinese postman problem. If there are m sections, m distinct routes must be found. Note that if desired, each section could be divided into two or three subsections and the optimal route calculated in each case. The driver could then pick which route to use contingent on how heavy a load he anticipates. In this case it would be necessary to find $2m$ or $3m$ distinct routes. The problem of finding the minimal route for a vehicle which cannot service the entire area in one trip due to a capacity constraint is called the m -postmen Chinese postman problem. Since in real life situations the vehicle may be constrained to begin and terminate each tour at some centralized location such as the disposal facility, each tour should in reality have a common origin. However, this added restriction will be ignored in this thesis due to the added complexity it brings to the problem. Obviously, finding routes for a fleet of vehicles falls in the category of the m -postmen Chinese postman

problem. Little if anything has been written in the literature concerning this problem.

Other factors must also be taken into account in routing trash collection vehicles besides the fact that a single vehicle cannot cover the entire city in one trip. Many cities contain one way streets, which unfortunately cannot be handled in the Chinese postman problem. Some work in the form of an existence theorem for an Euler tour, which is a path through a network which traverses each arc exactly once, has been done by Johnson^[13] on bidirected networks. A bidirected network, which is generally what must be used to model a real city street network, is a graph which contains both directed and undirected edges. Unfortunately, Johnson's work does not lend itself to determining an algorithm to find the minimum edge covering tour for a general bidirected network. However, experience has shown that one way streets in a real city are so sparse and are laid out in such a manner that once a solution is obtained for an identical network without the restriction of one way streets, it can be manipulated to conform to these added restrictions. It has even been proposed that collection vehicles be allowed to disregard one way street patterns since they travel so slowly. However, it is questionable whether the public can ever be made to accept this proposal.

Finally, an added complication in trash collection is that certain streets must be covered twice. While on quiet residential streets both sides can be picked up at once, this is not feasible on busy streets since it would have such a deleterious effect on traffic. Consequently, these streets must be covered twice. Fortunately, this may be easily

taken into account in the Chinese postman problem formulation simply by adding in extra arcs to the original network corresponding to those streets which must be covered twice.

2. Analysis of the Snow Plowing Problem

Snow plowing, like trash collection, is an important city service which requires a fleet of vehicles to cover every street. The obvious problems in snow plowing are parked cars, which prevent the plows from doing a thorough job, and stuck cars in the middle of the street, which prevent the plows from covering that street at all. Another major problem is routing the plows so they do their job thoroughly as well as efficiently. The strategy employed during snow storms in the city of Cambridge, Massachusetts is to call the plows out when there is a couple of inches of snow on the ground, and to keep them out until the storm is over. Basically, the city is divided into the same 75 sections as for trash collection, and each plow is assigned several sections. It may cover the sections any way it pleases, provided it plows the main streets first. Supervisors cruise the city and may rearrange assignments as they deem necessary to help out in trouble spots.

Some of the problems encountered in finding good routes for the plows are similar to those in trash collection. Since a single vehicle cannot cover the entire city, so that a fleet composed of m plows is required, the problem being faced is really the m -postmen Chinese postman problem. Also, since each street requires either two or four trips by a plow, depending on its width, the required number of edges must be

duplicated to allow for this. Finally, since most cities have some one way streets, the problem must be solved for a bidirected network rather than for an undirected network.

There is one additional restriction in snow plowing not present in trash collection. This is namely the restriction that main routes must be plowed first, connecting streets next, and residential streets last. The author knows of no efficient way to handle this constraint. One possibility is to place each street in a specific class, multiply its length by an appropriate weight, and use these weighted lengths to find the desired routes. The lengths of main streets, for example, may be multiplied by one, while minor streets might be multiplied by three. Thus, if certain edges must be covered more than once during the tour, this procedure would tend to favor the main streets. Another possibility would be to place each street in a specific class as before, and then to find the minimum edge covering tour for each class. Using this scheme, all main streets would be done before any connecting streets, which in turn would all be done before any residential streets.

A third, and probably the best procedure to handle this constraint is to find the minimum edge covering tour for the network while disregarding street priorities. Due to the richness in the number of alternate optimal tours, it will probably be possible to construct an optimal or perhaps near optimal tour which adheres to desired priorities.

CHAPTER III. HISTORICAL DEVELOPMENT OF SOLUTION TECHNIQUES FOR THE CHINESE POSTMAN PROBLEM

Although currently most real world edge covering problems are solved by trial and error techniques, some research has been conducted seeking more efficient algorithms. Unfortunately this research, conducted mainly in the area of graph theory, has not yet come up with an acceptable algorithm to solve a practical sized problem, or a problem with additional constraints such as limited vehicle capacity. Nonetheless, some of the work which has been done seems quite promising, and perhaps if combined with heuristic techniques can lead to at least improved, if not optimal, solution techniques for real world problems. With this in mind, descriptions of existing solution techniques for the Chinese postman problem are presented. At the conclusion of this chapter, a glossary of key terms presented is included as an aid for the reader.

1. Existence of an Euler Tour

The earliest mention of the minimum edge covering problem is by Euler in 1736. He proves theorems showing the existence of an Euler tour in either a directed or undirected graph. An Euler tour is a continuous path through a network such that each edge is covered exactly once. Obviously, it is the shortest possible tour capable of covering every edge. An Euler tour in an undirected graph exists if and only if the degree of every node is even. The degree of a node is the number of edges incident to it. If odd nodes exist, no Euler tour exists, so

certain edges must be covered twice if every edge is to be covered at least once. For a proof of this theorem, as well as an algorithm for constructing an Euler tour, see Appendix I. For a more complete discussion of Euler's work, as well as his existence theorem for an Euler tour in a directed graph, see Berge^[1].

2. Mei-Ko

The problem of what to do when odd nodes exist is taken up by Mei-Ko^[19]. He suggests that to minimize the distance required to cover every arc in a graph with $2n$ odd nodes, n paths must be duplicated between the odd nodes such that the total length of arcs travelled more than once is minimum. Every graph must have an even number of odd nodes because every arc has two end points, so the total degree of any graph is two times the number of arcs, which is an even number. A feasible solution is therefore n paths connecting the n pairs of odd nodes. The set of paths with the minimum total length is the optimal solution. To know when the optimal solution is achieved without enumerating every feasible set of paths, Mei-Ko proves the following theorem:

A feasible solution is optimal if and only if

- a) no arc is duplicated more than once;
- b) the length of added arcs on every cycle does not exceed half the length of the cycle. A cycle is a continuous path which starts and ends at the same node.

The necessity of this theorem is quite apparent. To prove necessity in part (a), note that removing two arcs duplicating the same original

arc does not change evenness since two edges are removed from each end point. For part (b), it should be apparent that removing all the added arcs, and then adding arcs instead where none were duplicated before, shortens the cycle while maintaining evenness. For a rigorous proof of this theorem, including its sufficiency, see Mei-Ko^[19].

Unfortunately, checking every cycle to see if the above theorem holds can be quite a chore in any reasonable sized network. Although some additional insight is gained towards the problem, no efficient algorithm is readily apparent.

3. Glover's Pseudo Edges

A quite different approach to the problem is developed by Glover^[10]. He poses the problem in the following way: given a graph G , determine a graph H which is a subset of G such that every node in the graph $G' = G+H$ is even. G corresponds to the original graph, while H corresponds to the set of duplicated arcs. Glover then proves certain theorems which lead to an algorithm to determine the optimal H , that is, the set of arcs with the minimum total length to be travelled twice. In order to understand Glover's work, some specialized notation must be explained first.

For every simple path (a path with no repeated edges) between nodes p and q in G , called Spq , a pseudo edge, Spq , can be associated between p and q , whose length is the negative of the length of Spq . Thus, between every pair of nodes p and q there can be many pseudo edges of varying lengths corresponding to the simple paths between p and q in G .

Glover's algorithm is basically quite simple. Select two nodes from the list of odd nodes, identify the shortest path between them, and remove those two nodes from the list of odd nodes. Next construct a new graph by removing the pseudo edges from the above shortest path, and adding pseudo edges corresponding to the true edges in the above shortest path. Continue until no odd nodes are left in the list. The resulting graph is the desired graph G' . For a more detailed description of the algorithm, along with several related remarks as well as a small sample problem, see Appendix II of this thesis.

In assessing the efficiency of Glover's algorithm, note that the major amount of time and effort is expended in determining m shortest paths between the m pairs of odd nodes, which would be required for a graph with $2m$ odd nodes. Determining the new graph from the old one by adding and subtracting pseudo edges after finding each shortest path is a simple bookkeeping operation requiring very little effort. Therefore, if an efficient shortest path algorithm can be found which is capable of handling pseudo edges, i.e., edges with negative length, this indeed would be an efficient way to solve the Chinese postman problem even for large networks. Dreyfus^[4] presents an excellent analysis of existing shortest path algorithms. In particular, he cites an algorithm by Yen^[23] requiring $N^{3/4}$ computations for a network with N nodes as holding particular promise for this problem. Future research in the adaptation of Yen's algorithm, possibly incorporating decomposition principles such as those proposed by Mills^[20], may prove quite rewarding.

A very important point to note is that at every step in the algorithm there exists the optimal solution for a graph whose odd nodes are only those already paired. Thus if several new odd nodes are added to the network, the algorithm can be resumed where it left off, and there is no need to start over again. A final point to observe is that in the application of the algorithm any pair of odd nodes remaining in the list can be chosen, which means that pairs can be chosen whose connecting shortest paths are obvious. This could prove of great computational value, especially in hand computations of problems with special structure where pairs of odd nodes may be more or less isolated.

4. Murty's Symmetric Assignment Problem

Another approach to the Chinese postman problem is taken by Murty^[21]. He proposes solving the minimum edge covering tour problem by solving a symmetric assignment problem whose cost matrix is made up of the lengths of the shortest paths between every pair of odd nodes. A feasible assignment is a set of cells in the cost matrix such that exactly one cell from each row and column of the cost matrix is in the assignment. Symmetric implies that if cell (i,j) of the cost matrix is included in the optimal assignment, then cell (j,i) must be included also. Diagonal entries would be infinity (or some appropriately large number) since going from node i to node i makes no sense for this problem. The reason that the assignment must be symmetric is that if a path connects node i to node j , then it obviously connects node j to node i . As has already been stated, Mei-Ko proves that the optimal solution to the Chinese postman problem

is a set of shortest paths connecting pairs of odd nodes, and whose total length is minimum. Murty offers a branch and bound algorithm for finding the minimal cost symmetric assignment, which for this problem is the set of shortest paths connecting every pair of odd nodes whose total length is minimum. The symmetric assignment corresponds to, therefore, those arcs which must be duplicated.

Before describing the algorithm, certain definitions must be given. A node N is a subset of K , the set of all feasible symmetric assignments, such that node N contains certain cells $(i_1, j_1)(j_1, i_1) \dots (i_r, j_r)(j_r, i_r)$ and does not contain cells $(m_1, p_1)(p_1, m_1) \dots (m_s, p_s)(p_s, m_s)$. For simplicity, node N is written as

$$N = \{(i_1, j_1)(j_1, i_1) \dots (i_r, j_r)(j_r, i_r); \overline{(m_1, p_1)} \overline{(p_1, m_1)} \dots \overline{(m_s, p_s)} \overline{(p_s, m_s)}\} .$$

An admissible unspecified cell at node N is any cell which is unspecified at node N and which does not lie in a row or column of a cell specified to be in that node. Branching from node N with the admissible unspecified cell (i, j) means to create from N two disjoint nodes N_1 and N_2 by in one case adding cells (i, j) and (j, i) , and in the other case adding cells $\overline{(i, j)}$ and $\overline{(j, i)}$. Using the earlier definition for N ,

$$N_1 = \{(i_1, j_1)(j_1, i_1) \dots (i_r, j_r)(j_r, i_r)(i, j)(j, i); \\ \overline{(m_1, p_1)} \overline{(p_1, m_1)} \dots \overline{(m_s, p_s)} \overline{(p_s, m_s)}\} \text{ and}$$

$$N_2 = \{(i_1, j_1)(j_1, i_1) \dots (i_r, j_r)(j_r, i_r); \\ \overline{(m_1, p_1)} \overline{(p_1, m_1)} \dots \overline{(m_s, p_s)} \overline{(p_s, m_s)} \overline{(i, j)} \overline{(j, i)}\} .$$

An important part of the algorithm requires reducing a matrix C to obtain a bound at each node. Basically, this amounts to subtracting constants from the rows and columns of C so as to get at least one zero in each row and column of the matrix. If done in a specific way, the optimal assignment is contained among the zeros of the reduced matrix, and a cost known as the reduction of matrix C is obtained in the process. The reduction is the lower bound at that node, and may even be the optimal answer to the problem. Murty suggests using the Hungarian method to reduce matrices. This method, using an algorithm originally developed by Kuhn^[14,15], may be found in Appendix III of this thesis.

Using the above definitions, the following is the basic framework of Murty's branch and bound algorithm.

Stage 1: Find a lower bound on the set of feasible symmetric assignments, written $LB(K)$, by finding the reduction of the original cost matrix C_0 . The algorithm terminates if there exists an optimal assignment which is symmetric. Otherwise branch from K .

General Stage m : K has been partitioned into several nodes, and those nodes not currently used for branching are the terminal nodes at this stage. A lower bound has been calculated on the cost of the minimal symmetric assignment at each node, and the terminal node with the smallest lower bound is the minimal terminal node at this stage.

Optimality Criterion: The algorithm is terminated whenever there exists a minimal terminal node which contains only one symmetric assignment, or when there exists a minimal terminal node in which the optimal assignment is a symmetric assignment. Convergence is guaranteed since eventually

every feasible set of symmetric assignments will be generated among the various terminal nodes.

If the optimality criterion is not satisfied at stage m , then certain involved branching rules must be used to select which node to branch from. These rules, which may be found in Appendix III, attempt to reach a feasible solution as quickly as possible while at the same time maintaining the lowest bounds possible. It is this author's opinion that a more efficient set of branching rules can be achieved for the highly structured symmetric cost matrix of the Chinese postman problem by concentrating more on finding a feasible solution as quickly as possible, rather than maintaining the smallest possible bounds. For more on this subject, see Chapter IV, section 1, of this thesis.

Regardless of what branching rules are used, once the optimality criterion is satisfied, the optimal assignment in the minimal terminal node at the final stage is the optimal symmetric assignment to the problem. As discussed earlier, by using as costs the shortest paths between pairs of odd nodes and making the diagonal elements very large, the optimal assignment shows which paths must be added to the original network to obtain an optimal graph on which to perform an Euler tour.

Computational considerations on Murty's algorithm are delayed until the next section where Edmonds' work is described, since both authors take a similar approach to the problem.

5. Edmonds' Matching Algorithm

Edmonds [5,6,7,8], in his early work, proposes a scheme similar to

Murty's to solve the Chinese postman problem. Shortest paths between every pair of odd nodes must be found, and Edmonds develops an algorithm to find the optimal assignment using graph theory techniques. He translates the cost matrix into a graph with one node for each odd node, and edges representing the shortest paths between odd nodes in the original network. He then proposes finding the optimal perfect matching for this graph, where a perfect matching is a subset of edges of the graph such that each node is incident to exactly one edge.

Edmonds' algorithm is really a dual algorithm. Weights are assigned to each node, and complementary slackness is used to show that for any edge in a perfect matching, the sum of the weights assigned to its end points plus certain additionally defined quantities called set weights must equal the cost of the edge, which in this case corresponds to the length of a shortest path. The algorithm begins by assigning any feasible weight to each node, and then proceeds by improving on these initial choices until optimality is achieved. Unfortunately, since the node weights will in general not be integer, a rather involved process is required to update node weights. While each individual step in the algorithm may be quite simple, there are several alternatives which must be checked for after each step, a great deal of bookkeeping is involved, and a great many iterations of the basic steps may be required to obtain the optimal solution. For a complete description of the matching algorithm, see Edmonds [8].

The approach to the Chinese postman problem taken by Murty and Edmonds, mainly that of finding the shortest path between every pair of

odd nodes and then finding the minimum pairing so that every odd node is paired to exactly one other odd node, is certainly a very natural one. Without computational experience, it is difficult to determine which algorithm is the better one. Unfortunately, though, this general approach to the problem has a very serious shortcoming, namely the problem of obtaining the original cost matrix of shortest paths. For a network with n odd nodes, $n^2 - n$ shortest paths must be found. If no one way streets or other restrictions are imposed, so that distances between nodes are the same in either direction, then the matrix is symmetric and half that number of calculations must be made. Even so, the number of shortest paths which must be found grows quite rapidly, approximately with the square of the number of odd nodes. Compare this with Glover's algorithm where the number of shortest paths to be determined varies linearly with the number of odd nodes.

However, the situation is not as hopeless as it appears at first glance. Dreyfus^[4] references several algorithms capable of finding shortest paths between all pairs of nodes in a network. Since the Chinese postman problem requires only the shortest paths between all pairs of odd nodes, these algorithms would have to be modified somewhat. For a network with N nodes, Floyd's^[9] nine line ALGOL algorithm requires $N(N-1)(N-2)$ computations to determine the shortest paths between all pairs of nodes. This would appear to be as efficient as possible in light of the present state of the art, and is capable of handling moderate sized networks. To handle large networks representing modern cities, some type of decomposition such as described by Mills^[20] would

probably be necessary. Still, a great deal of computation is required before either Murty's or Edmonds' algorithm can be applied, which is certainly a serious drawback.

Edmonds is currently extending his matching algorithm to solve what he calls the b-matching problem. This problem is like the perfect matching problem, except that instead of exactly one edge in the matching meeting each node, an upper and lower bound can be placed on the number of edges in the matching incident to any particular node. Although this algorithm is reportedly capable of efficiently solving the Chinese postman problem, exactly how it is applied and specific details on the algorithm itself are unknown since no published material has appeared yet.

6. Johnson's Existence Theorem for an Euler Tour on a Bidirected Graph

So far all the work reported on has been concerned only with the Chinese postman problem on an undirected network. Since most cities have some one way streets, what is really required is a technique for bidirected networks, i.e., graphs with both directed and undirected arcs. Unfortunately, very little has been written on this subject. Johnson^[13] in a recent paper proves an existence theorem for an Euler tour on a bidirected network.

Johnson shows that an Euler tour exists in a connected bidirected graph if and only if:

- 1) Every node has even degree; that is, if the number of directed arcs leaving the node minus the number of directed arcs entering the node plus the number of undirected arcs incident to the node is even;

2) For every proper subset M of nodes N, the degree of all nodes in M due to directed edges (the net number of edges leaving M) must be less than or equal to the number of undirected edges with one end in M and the other end in N-M.

Intuitively, the necessity of this theorem is quite obvious. Condition (1) is necessary if it is always to be possible to leave a node once it is entered. Condition (2) states that the net number of directed edges leaving any subset of nodes must be less than or equal to the number of undirected edges with exactly one end in that subset to insure being able to return to it every time after leaving it. A proof for the sufficiency of these two conditions, which requires induction, is presented by Johnson^[13].

As part of the proof of this existence theorem, Johnson constructs the following network flow problem:

$$\sum_m (X_{nm} - X_{mn}) = -\delta_n(D) \quad \text{for } n \in N \quad (5)$$

$$X_{mn} = 0 \text{ or } 1 \quad \text{for } n \neq m \quad (6)$$

where

$X_{mn} = 1$ if the undirected edge from node m to node n is traversed in that direction, and 0 otherwise

$\delta_n(D) =$ the degree of node n due to the directed edges incident to it

Equation (5) insures continuity of flow through the network, and equation (6) restricts X_{mn} to being a zero/one variable. If a feasible solution exists to this problem, then an Euler tour exists for the bidirected network.

It is fortunate that it is only necessary to prove the existence of a solution to the above network flow problem, since checking condition (2) for every possible subset of nodes could prove to be a very difficult task. Unfortunately, this existence theorem represents the present state of knowledge for finding the minimum tour on a bidirected graph. Much research remains to be done in techniques which can be used when an Euler tour does not exist on the given bidirected network. Experience has shown, though, that for most cities, if an Euler network is constructed without regard to one way streets, an Euler tour can usually be drawn which adheres to these additional constraints.

7. Marks and Liebman's Travelling Salesman Approach

Although this thesis is primarily concerned with continuous public sector vehicle routing problems, the following approach developed by Marks and Liebman^[17] for the discrete private sector case is presented since it is the only algorithm appearing in the literature which makes any attempt at handling the capacity constraints present in real world problems. Briefly, the travelling salesman problem is the problem of finding the minimum tour for a salesman who must visit N cities, with the distance from each city to every other city known. While formulating this problem is extremely simple, solving it for any reasonably sized problem has proven to be quite difficult due to the extremely large number of feasible solutions. A great deal has been written about this problem in the literature due to its wide applicability, and for a concise review of what has been done, see Marks and Liebman^[17].

An interesting extension of the travelling salesman problem proposed by Liebman and Marks is finding the minimum tour for a salesman who can only visit K cities at a time before returning to his base location, where K is assumed less than N . Thus it is necessary to find $m = K/N$ distinct tours. This problem is referred to as the m -salesman travelling salesman problem. The algorithm for it proposed by Marks and Liebman involves setting up a less constrained problem which can be solved by using a general network flow algorithm, the out-of-kilter algorithm. If the solution to the less constrained network problem is feasible for the m -salesman travelling salesman problem, then the algorithm terminates. If not, a branch and bound procedure must be performed until the optimal solution is obtained. For a more detailed description of the algorithm, see Marks and Liebman^[17].

It is interesting to note how the m -salesman travelling salesman problem formulation can be applied to the problem of trash collection. Marks and Liebman propose dividing the network to be covered into small collection areas, each of which is considered a city in the formulation. Distances are measured from centroid to centroid. Each truck can visit K areas before it becomes full and must empty its load.

This formulation is more realistic than the Chinese postman problem formulation in the sense that it recognizes a capacity constraint which prevents a single vehicle from covering the entire network in one tour. However, it also has a drawback since it requires the treatment of what is a continuous problem in the public sector as a discrete problem for the sake of solution. Once the order in which the areas are to be visited

is determined, there remains the problem of determining in what order the streets within each area will be serviced. Unfortunately, the authors have little to say on this problem. One way to get around this difficulty would be to treat the midpoint of each street as a city. However, one readily sees that this is an unrealistic approach in light of existing techniques, since it would involve first of all determining thousands of shortest routes between midpoints of streets to obtain a data base for the algorithm, and secondly no existing travelling salesman type algorithm could handle anywhere near the resulting number of cities.

Computational experience with the Marks and Liebman algorithm bears this out. While encouraging for small problems, it appears hopeless even for modest sized problems. An optimal solution for an 8 city problem with $m=2$ was found in .03 minutes on an IBM 7094 computer. However, for a problem just twice this size with 16 cities and $m=4$, no feasible solution was even found after 2.12 minutes due to lack of storage. While more efficient programming may improve this algorithm somewhat, it would appear that a radical change is necessary to make it a useful tool for real problems.

8. Glossary

Although most terms are defined in the body of the text where they first appear, the following glossary is presented as an aid to the reader in looking up subsequent referrals to these terms.

Bidirected graph - a graph made up of both directed and undirected edges.

Chinese postman problem - the problem of finding the minimum edge covering tour on an undirected graph.

Cycle - a closed continuous path which both begins and terminates at the same node.

Degree of a node - the number of undirected edges incident to a node plus the number of directed edges leaving that node minus the number of directed edges entering that node.

Directed edge - an edge which must be traversed in a specific direction.

Directed graph - a graph made up of directed edges only.

Edge covering tour - a path through a network such that each edge is covered at least once.

Euler network - an undirected network such that the degree of every node is even.

Euler tour - a path through a network such that each edge is covered exactly once.

Even graph - an Euler network.

Graph - a set of nodes plus a set of edges connecting them. The words network and graph are used interchangeably in this thesis.

m-postmen Chinese postman problem - the problem of finding m distinct tours on an undirected network such that every edge is covered at least once and the total distance travelled is minimum.

Network - see graph.

Path - a set of edges in an undirected graph such that every edge terminates at the node where the following edge begins.

Pseudo edge - an edge associated with any elementary path in a network (a path which touches any node or arc once at most) whose length equals the negative of the length of its associated path of real edges.

Undirected edge - an edge which may be traversed in either direction.

Undirected graph - a graph made up of undirected edges only.

CHAPTER IV. PROPOSED NEW SOLUTION TECHNIQUES

1. Modification of Murty's Assignment Algorithm

Murty and Edmonds both propose solving the Chinese postman problem by finding the shortest path between every possible pair of odd nodes, and then selecting the optimal pairing. Here optimal pairing implies that every odd node is paired to exactly one other odd node by their shortest connecting path so that the total length of selected paths is minimum. Both authors ignore the initial problem of determining all the shortest paths, and go directly to the problem of pairing nodes. Unfortunately, since neither author offers any computational experience, it is difficult to determine which algorithm is the better one. Edmonds' algorithm, although basically simple to understand, may be computationally arduous to carry out, and requires great amounts of bookkeeping. Murty's algorithm, being a branch and bound algorithm, is data dependent and may require either very many or relatively few branches to achieve optimality. While only computational experience can determine which algorithm has the edge, this author proposes a third scheme to be considered which is a modification of Murty's algorithm and may be advantageous for the highly structured and symmetric cost matrix of the Chinese postman problem.

Having obtained the starting cost matrix, the problem is to determine an optimal symmetric assignment on it. The problem is rather specialized since all diagonal entries are infinity, and the cost matrix will generally be symmetric. In a situation such as this, one might be tempted to assume that solving a regular assignment problem using the Hungarian method will

necessarily result in a symmetric optimal assignment. Unfortunately, this is not so, as can be seen from the following counterexample in Figure 4-1.

∞	10	1	10	10	1
10	∞	10	1	1	10
1	10	∞	10	10	1
10	1	10	∞	1	10
10	1	10	1	∞	10
1	10	1	10	10	∞

Figure 4-1. Example of a Symmetric Cost Matrix with Unsymmetric Optimal Assignment

Two alternate, but not symmetric, optimal assignments exist;

$$a_1 = \{(1,3)(2,4)(3,6)(4,5)(5,2)(6,1)\} \text{ and}$$

$$a_2 = \{(3,1)(4,2)(6,3)(5,4)(2,5)(1,6)\}.$$

Even though a symmetric assignment is not guaranteed in such a case, there is an excellent chance that one will result due to the symmetry of the situation. If a symmetric assignment does not result on the first branch, it may be possible by examining the assignment obtained to choose cells to branch from so that a symmetric assignment is quickly obtained. For instance, if a reduction indicates a symmetric assignment except for two rows and columns, branch from among the cells necessary to complete the symmetric assignment, even if they are not the cells specified by

Murty's algorithm. Though sacrificing something in determining bounds, the important thing in any branch and bound algorithm is to obtain a good feasible solution as quickly as possible, which this approach tends to produce.

In other situations, where the original cost matrix is not symmetric, a more efficient strategy may result by starting off with reducing matrix C' , where $C' = \frac{C+C^T}{2}$, and hence differs from the original matrix C , since it has been made symmetric. This would increase the probability of obtaining the optimal symmetric assignment initially, and would cut in half the number of times the Hungarian method need be applied at each branch because of the enforced symmetry. Although the costs of all symmetric assignments would remain unchanged, this is not true of non-symmetric assignments. The main drawback of this method would be that if a reduction did not result in a symmetric assignment, the lower bound obtained would probably not be as good as if the reduction were performed on the original matrix. In the long run, it would appear more advantageous to perform fewer calculations to determine bounds, and to attempt to artificially obtain symmetric assignments, at the expense of accepting poorer bounds. Experience with branch and bound algorithms on highly structured problems indicates that it is usually better to solve a more involved subproblem fewer times, than to solve a simpler problem many more times. Certainly a small amount of art is involved in this approach, and a great deal of practical experimentation is necessary before it can be known for certain whether it does indeed offer any advantages. Moreover, it is doubtful that this approach will prove efficient for the

Chinese postman problem due to the large size of the original cost matrix and the amount of computation required just to obtain it before beginning to find the optimal symmetric assignment. Intuitively it appears wasteful to calculate distances between nodes which will not be paired in the optimal solution.

2. Decomposition Algorithm

Although there are algorithms to solve the Chinese postman problem which are efficient for modest sized problems, no good algorithm has been presented in the literature thus far which is capable of solving the problems encountered in any reasonable sized city with thousands of streets. The reason is that computational difficulty increases exponentially rather than just linearly with the number of nodes and arcs in the network. This is due to the fact that as the number of streets grows, the number of feasible solutions grows even more rapidly, making it extremely difficult to determine the optimal solution.

One obvious possibility in a situation like this is to investigate the use of decomposition. What is required is a procedure which would allow one to break a large network up into small sections, solve the Chinese postman problem for each section, and then recombine all the sections so that the optimal solution to the original network is easily obtainable. Unfortunately, such a procedure is not obvious for this problem since there is no way of knowing (without first solving the problem) whether or not two odd nodes which should be connected by a shortest path in the optimal solution are being put into two separate

sections. The reason for this is that the optimal pairing of odd nodes is generally dependent on the structure of the entire network rather than just localized sections.

However, in some situations, a heuristic decomposition may be desirable, possibly because the incremental gain from obtaining the optimal solution is not worth the added computational effort, or perhaps because it is not even possible to obtain the optimal solution for the entire network due to computational limitations. This would be especially true of a large symmetric network associated with a well planned city with regular streets. In such a case there would be relatively few odd nodes, and they would be fairly well localized, so with a little planning the network could be divided into small sections, with the center of each section containing a group of odd nodes. It is desirable to keep the odd nodes within the center of each section since this minimizes the probability of placing two nodes in separate sections which should be paired in the optimal solution.

The following set of rules is a guide for partitioning a network, along with a procedure for improving the feasible solution obtained by combining the optimal solutions from each section.

Decomposition Algorithm

- (1) Partition the set of nodes of the network into the desired number of mutually exclusive, collectively exhaustive subsets such that each subset contains an even number of odd nodes. Every edge with both end points in the subset is included in it. Make the partitioning as natural as possible in that each subset with its associated

edges should form a connected graph with the odd nodes occupying as compact and centralized a location as possible.

- (2) Call those edges in the original network whose end points are contained in different subsets boundary edges. Add to each subset its associated boundary edges, and hence each boundary edge appears in two different sections. Also include in each section edges of the shortest paths from other sections which join the end points of its boundary edges. This in effect adds the shortest path incident to nodes outside the subset between pairs of nodes on the perimeter of the subset, i.e., incident to boundary edges. These boundary edges along with the shortest paths connecting them form sets of edges throughout the network similar in concept to the barrier sets first presented by Mills^[20].
- (3) Solve the Chinese postman problem for each section by determining which edges must be duplicated. Add pseudo edges (edges whose lengths are the negative of their physical lengths) corresponding to the duplicated edges.
- (4) Recombine the sections, adding any edges appearing in more than one section once as a real edge plus as many times as a pseudo edge as the number of sections in which it appears as a pseudo edge.
- (5) Check for cycles with negative length. If any occur, remove the pseudo edges of the cycle, and then add pseudo edges corresponding to the true edges of the cycle. Repeat until no negative cycles can be found.
- (6) The optimal solution is obtained by adding to the original network those edges corresponding to remaining pseudo edges.

Justification of the Algorithm

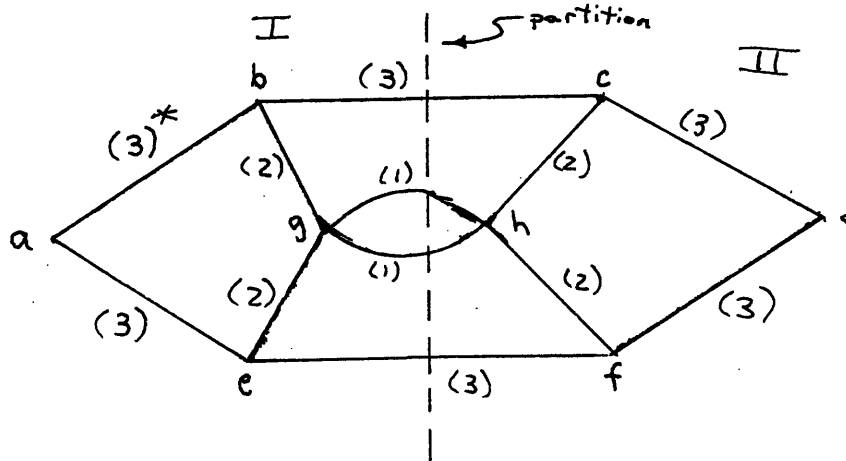
Note that this algorithm partitions the original set of nodes into mutually exclusive sets. Edges are assigned to each set so that the shortest possible path between every pair of nodes in the set is present, and hence edges incident to nodes on the boundary of a set are included in more than one set. The assignment of edges to more than one section is necessary to ensure that the minimum set of shortest paths connecting odd nodes in the current portion of the problem has been found. This concept is similar to the barrier sets Mills^[20] constructs in his decomposition algorithm for shortest path problems. In real road networks where the shortest path between two points is a straight line, finding the shortest path between end points of boundary edges merely means including the street connecting them.

Of course, merely combining solutions to the subproblems does not generally guarantee the optimal solution for the original network, which is why step (5) must be included. These rules use the criterion for optimality first presented by Mei-Ko^[19]. Glover^[10] suggests a similar procedure whereby one starts with any feasible solution and attempts to improve it, but quickly rejects it as requiring too much computational effort to determine negative cycles. Nonetheless, it should not be too hard to visualize the ease with which negative cycles might be found in a network resulting from a city laid out on a square grid pattern with necessarily relatively few odd nodes, and hence a simple pattern of pseudo edges. Computational experience with an old, rather poorly laid out city, which is discussed in Chapter V, bears out this assumption.

Certain computational procedures which are described turn the problem of finding negative cycles into a pattern recognition problem, for which the human mind is especially well suited, but which remains beyond the grasp of computers.

The search for negative cycles is simplified by starting with a good feasible solution. By judiciously partitioning the network in the first place, the feasible solution obtained by combining the optimal solutions for each section should be relatively good. Thus, one is in a good starting position to seek the optimal solution for the whole network since few negative cycles should exist. An already feasible solution is always being improved upon, so even if the computation is stopped before achieving optimality, a good feasible solution is available.

The following short example in Figure 4-2 is presented to quickly



Odd nodes = {b,c,e,f}

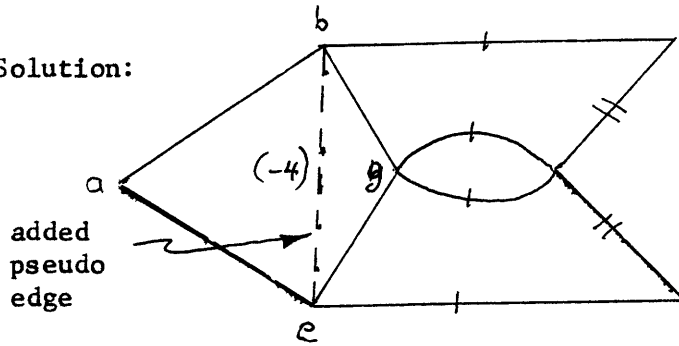
Section I = {a,b,g,e}

Section II = {c,f,h,d}

*Numbers in parenthesis by each edge indicates its length

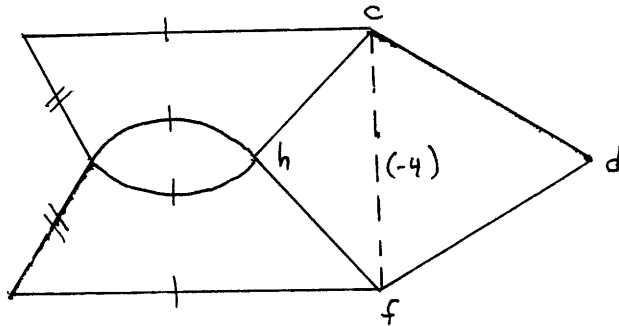
Figure 4-2.a. Example of Decomposition Algorithm

Section I Solution:

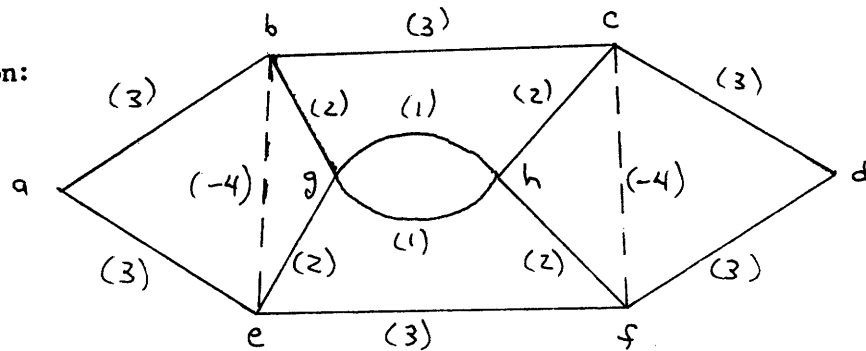


- a. Single slash denotes boundary edge
- b. Double slash denotes shortest path joining boundary edges

Section II Solution:

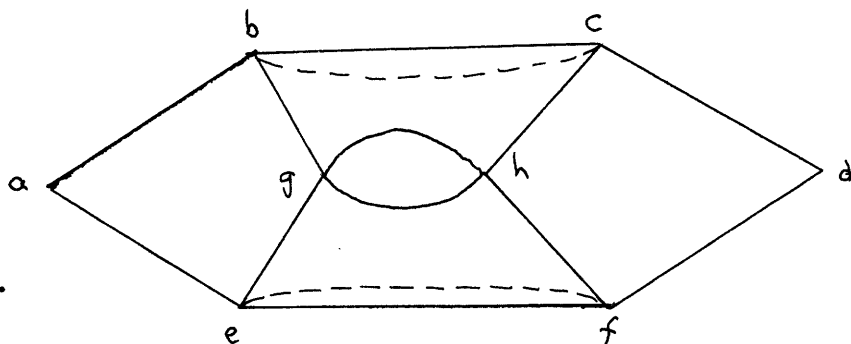


Combined Solution:



Cycle b,c,f,e is negative.

Optimal Solution:



Duplicate edges (b,c) and (e,f).

Figure 4-2.b.

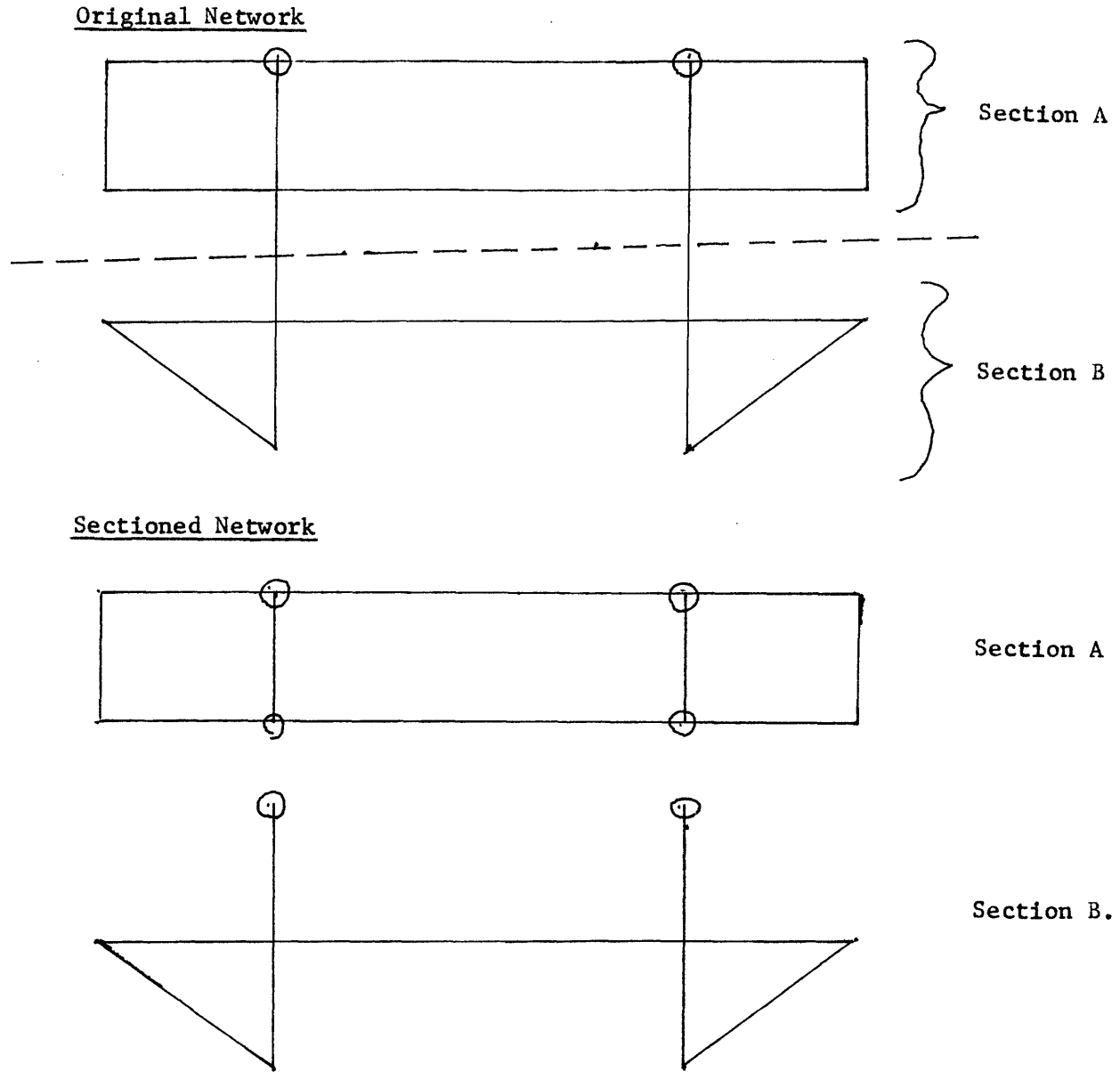
demonstrate this technique. Solution of a large size problem along with computational experience, as well as remarks concerning helpful computational techniques, are deferred until Chapter V.

3. Solution of m-Postmen Chinese Postman Problem

While solution techniques do exist for finding the minimal route for a single vehicle which must cover the entire network, this in general is not exactly the problem which planners face. The more practical problem, which is called the m-postmen Chinese postman problem, involves finding the shortest route for a vehicle under some additional capacity constraint, requiring it to return to a base location after travelling at most a specified distance, before it can resume its tour.

With what has already been presented, it is not clear how to handle this extension to the problem. One ad hoc approach to this problem is to divide the original network up into sections, the size specified by the capacity constraint, and then to find the minimal edge covering tour for each section. This is certainly not a very good approach, as can be seen from Figure 4-3. If one just blindly divides up the network, what is an even node in the original network will often be turned into odd nodes in each of two sections. This is not desirable since as has already been pointed out, odd nodes mean additional duplicated arcs.

The key to a good solution of the problem is to divide the network so as not to create odd nodes from even ones. Fortunately, it is easy to adhere to this constraint by ensuring that each section has a continuous boundary. Group adjacent edges incident to any one node into



Note - There are only two odd nodes (indicated by circles) in the original network, but there are six in the two resulting sections.

Figure 4-3. Ad Hoc Partitioning of a Network

pairs, and always make assignments by pairs to any particular section. If it is assumed that an Euler network is present to begin with, so that the degree of every node is even, it will always be possible to make these assignments in pairs, since exactly two edges incident to a node must form the boundary of each of its adjacent sections. By starting with an even number of edges incident to the node, an even number of edges will always remain to be included with adjacent sections. Obviously, any particular node can only be on the boundary of as many sections as half the number of edges incident to it, since two edges must form the boundary of each such section.

Algorithm for m-Postmen Chinese Postman Problem

- (1) Using one of the Chinese postman problem algorithms, create an Euler network from the given network, so that every node is even.
- (2) Arbitrarily choose loose boundaries in accordance with capacity constraints.
- (3) Construct the optimal boundary by ensuring that each section has a continuous boundary so that an even number of edges is incident to every node.

Example

We are given the following loosely partitioned Euler network shown in Figure 4-4.

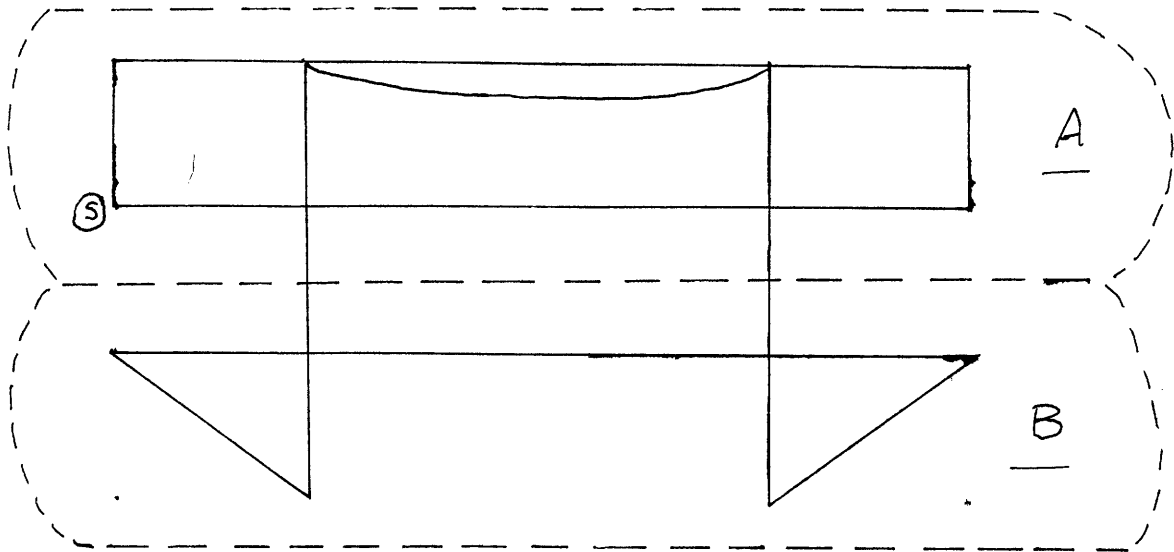


Figure 4-4. m-Postmen Example

We arbitrarily start at (S), and assign edges to form the boundary of section A.

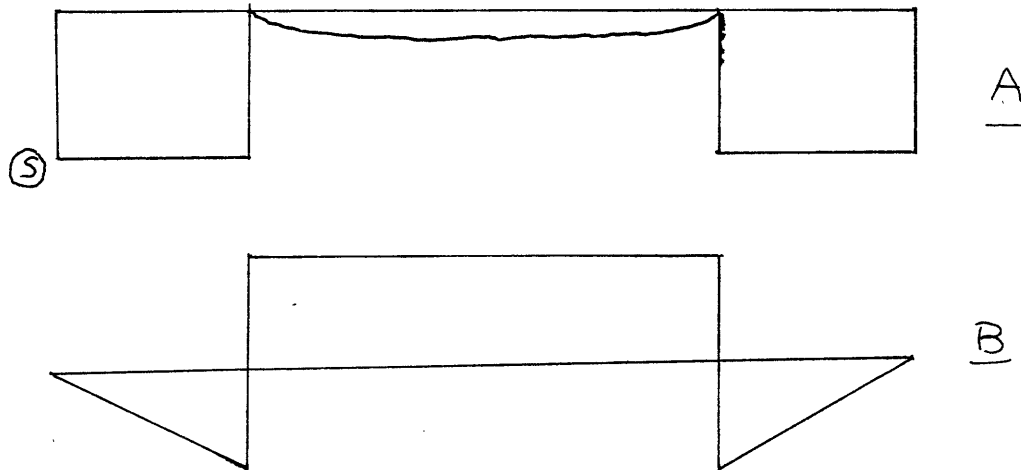


Figure 4-5. Solution of m-Postmen Example

Note that both A and B are Euler networks, as is expected from following the algorithm, so it is possible to proceed directly to find their respective minimal edge covering tours.

Unfortunately, this technique has two obvious shortcomings. First, since the original loose partitioning as well as the final optimal boundary is so arbitrary, some trial and error will be involved in defining sections closest to the size specified by the capacity constraint. Second, this technique does not take into account getting from the base location to each section. While these are certainly very real problems, it may still be appropriate to apply this algorithm. For one thing, the capacity constraint is often a rather loose bound, so that the variation in size of sections is not significant. As far as the division of the network not being optimal with respect to getting from the base location to each section, this is probably a second order effect because major arteries would probably exist to make such travel time minimal. It is reasonable to assume that each section should cover a contiguous area so as to avoid confusion for drivers, and so that the public isn't overly annoyed by noisy trucks continually criss-crossing quiet residential streets.

This algorithm is one of the first attempts to handle analytically the capacity constraints present in many vehicle routing problems. As with many such early efforts, it is not intended to be a final solution to the problem, but rather a first step in the search for an acceptable solution. Even so, considering the computational ease of the algorithm, as well as the fact that traditional routings used in many cities are in obvious need of improvement, the use of this algorithm can be justified in such cases until a more efficient analytic technique is developed.

CHAPTER V. COMPUTATIONAL EXPERIENCE ON A LARGE SCALE PROBLEM

Due to the basic simplicity of the decomposition algorithm presented in Chapter IV, section 2, of this thesis, the decision was made to ascertain its computational efficiency by attempting to solve a life size problem with it. The city selected was Cambridge, Massachusetts, since it represents a relatively large, very old, difficult to analyze city, and also due to its proximity to the author which made it convenient to obtain data.

1. Problem Description

The city of Cambridge was founded in 1630. As can be imagined, many of its streets are narrow and winding, with many intersections composed of anywhere from three to seven streets coming together at odd angles. It is anything but a well laid out city, and hence can be considered an appropriate test for the algorithm.

The public works department divides the city into five sections for the purpose of trash collection. One section is serviced each weekday. It was decided to work this algorithm on Tuesday's section, which is in the southwest corner of the city, and composed of relatively short residential streets with many odd intersections. Altogether, this area contains about 250 streets and 144 odd nodes, not including dead end streets. Dead end streets are ignored in the computations since they obviously must be covered when the streets they are off of are covered. Figure 5-1

is a map of Cambridge, with the darkened area indicating the section under consideration. Figure 5-2 is a detailed map of the section.

2. Computational Procedure

The first step according to the algorithm is to partition the original set of nodes into the desired number of mutually exclusive subsets, which in this case was decided to be three. However, before doing this it is helpful, for bookkeeping purposes, to mark each odd node with a small colored dot. This makes the odd nodes stand out, not only to be sure of including an even number of them in each subset, but also so that localized groups of them may be spotted and placed in the same subset. Whenever possible, choose nodes falling on long straight streets to form the boundaries of the subsets, and also avoid choosing an odd node as a boundary node. Such choices make finding shortest paths between boundary edges easier, and minimize the probability of placing into separate sections odd nodes which should be paired. Figure 5-3 presents one partitioning of the network. Figures 5-4, 5-5, and 5-6 are maps of sections I, II, and III respectively, including required boundary edges and shortest paths.

The next step is to solve the Chinese postman problem for each of the three subsets. Even though each subset contains approximately 50 odd nodes, the author had extremely encouraging results using trial and error to pair nodes and then making improvements by eliminating negative cycles. Section I took almost 60 minutes to solve, while section II took 30 minutes, and section III less than 20 minutes. This is rather good

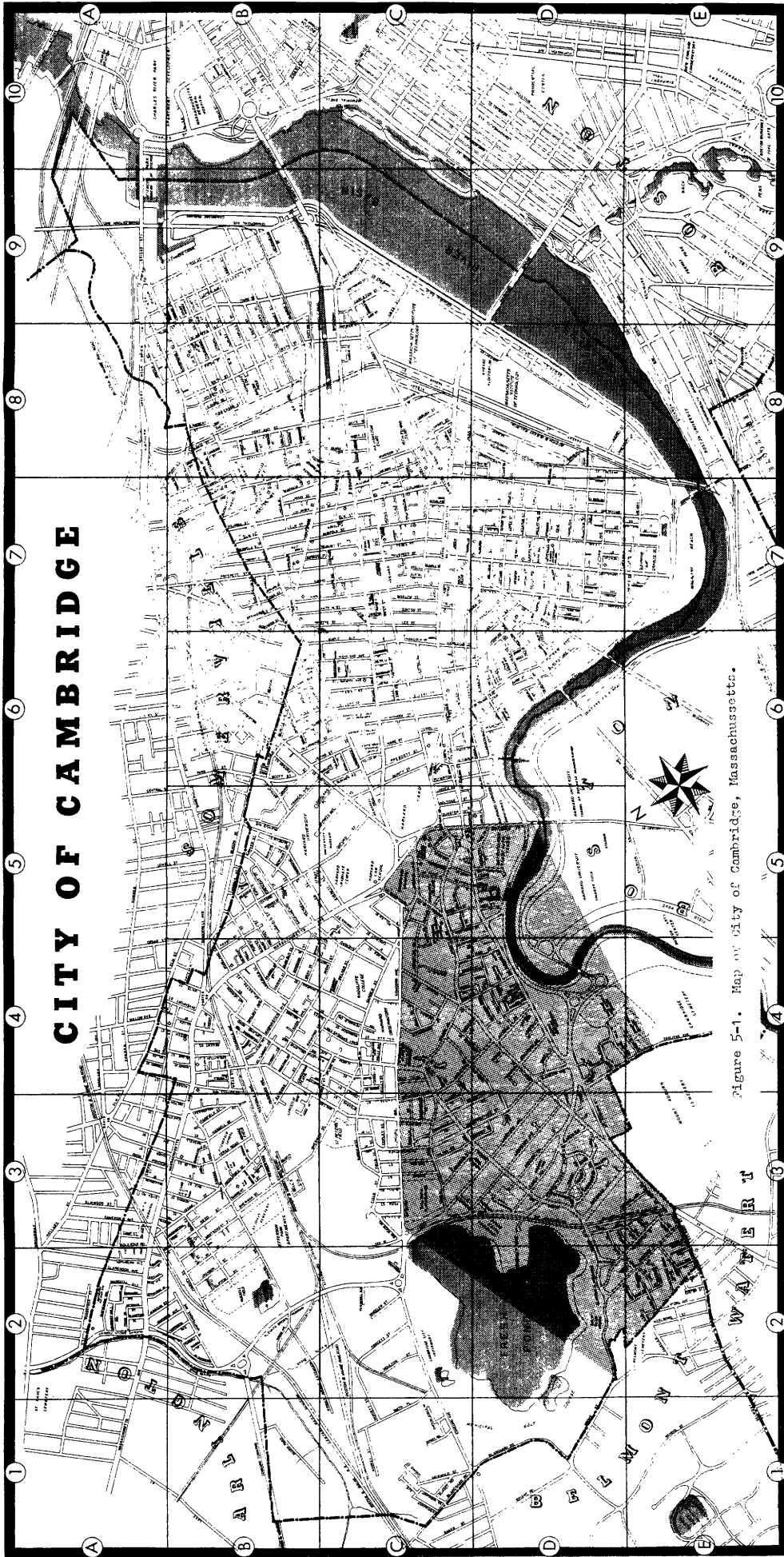


Figure 5-1. Map of City of Cambridge, Massachusetts.

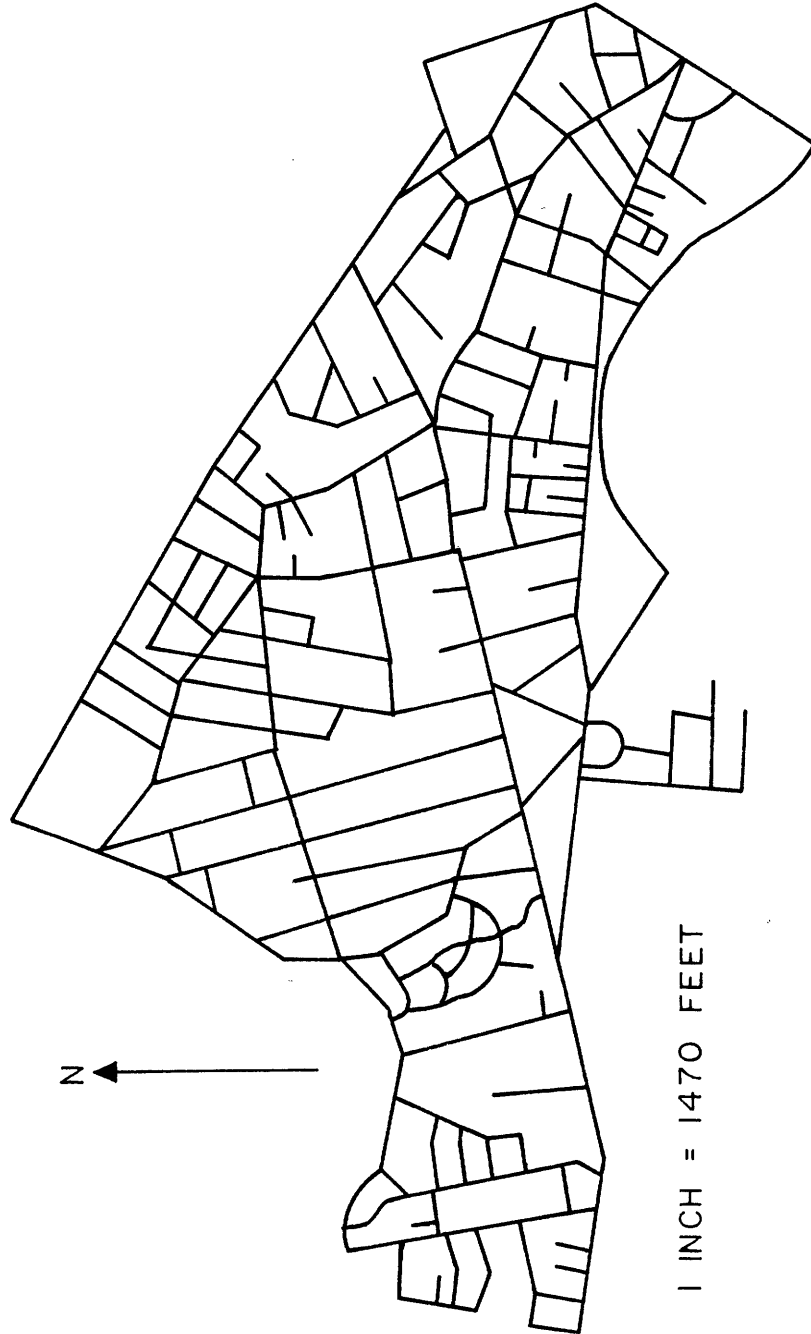


Figure 5-2. Map of Southwest Portion of Cambridge, Massachusetts.

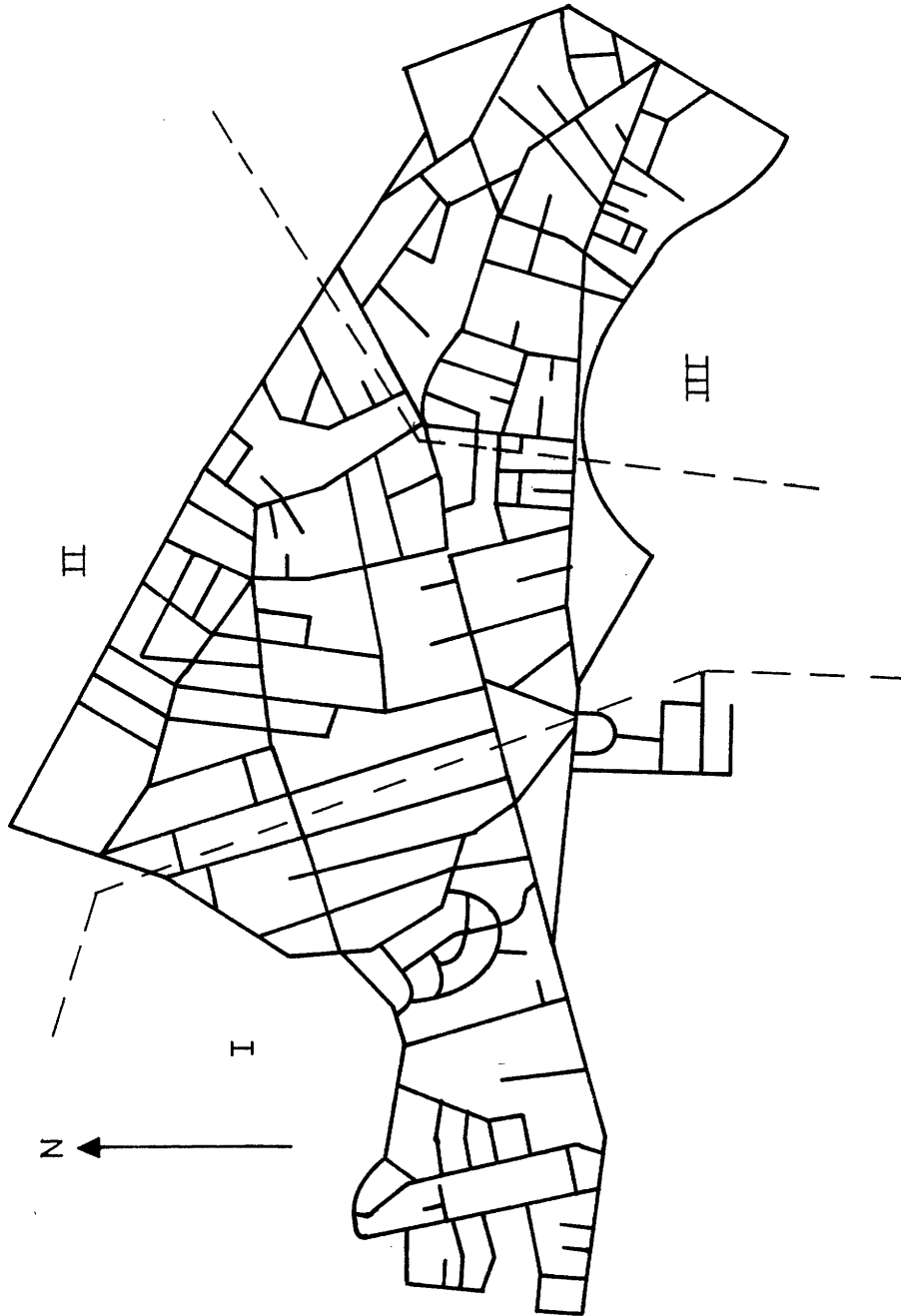


Figure 5-3. Map Showing Partitioning of the Network Into Three Sections.

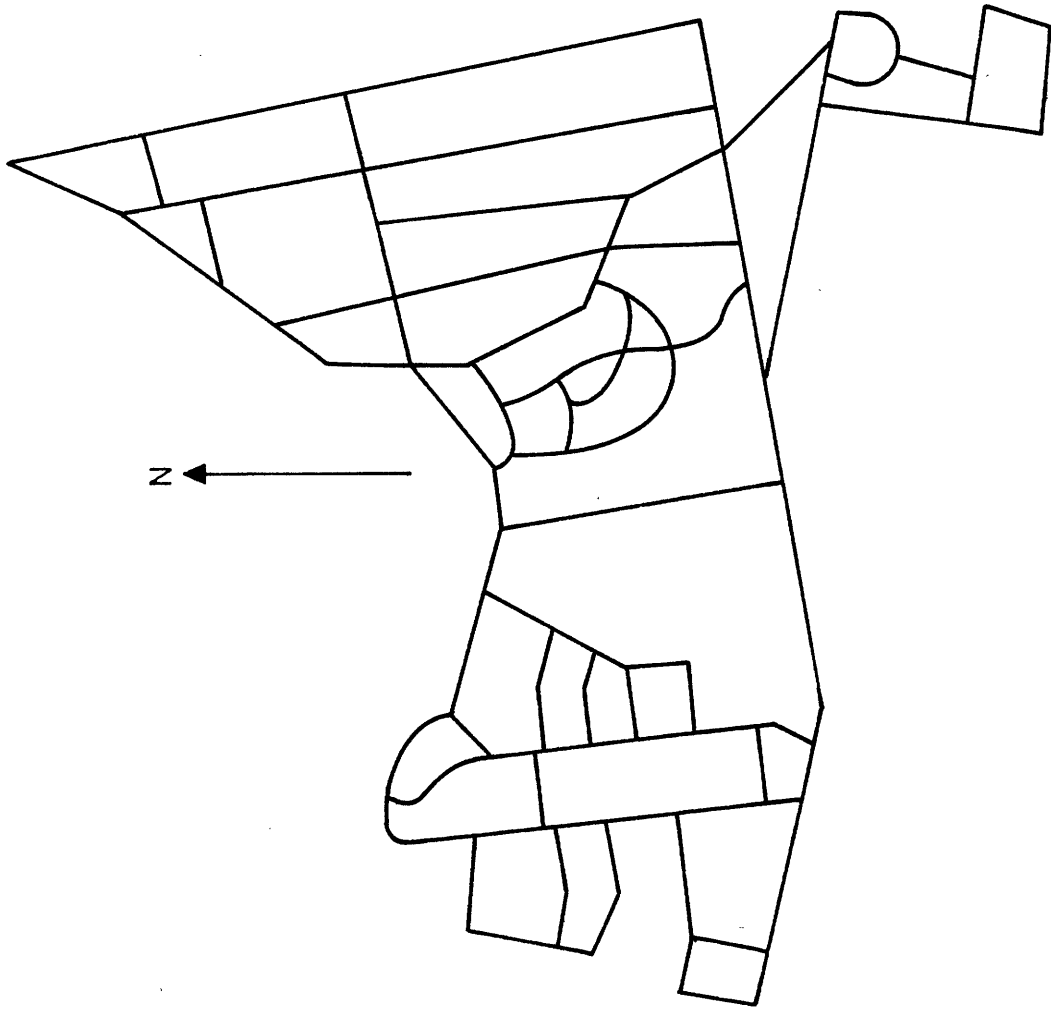


Figure 5-4. Map of Section I Including Boundary Edges.

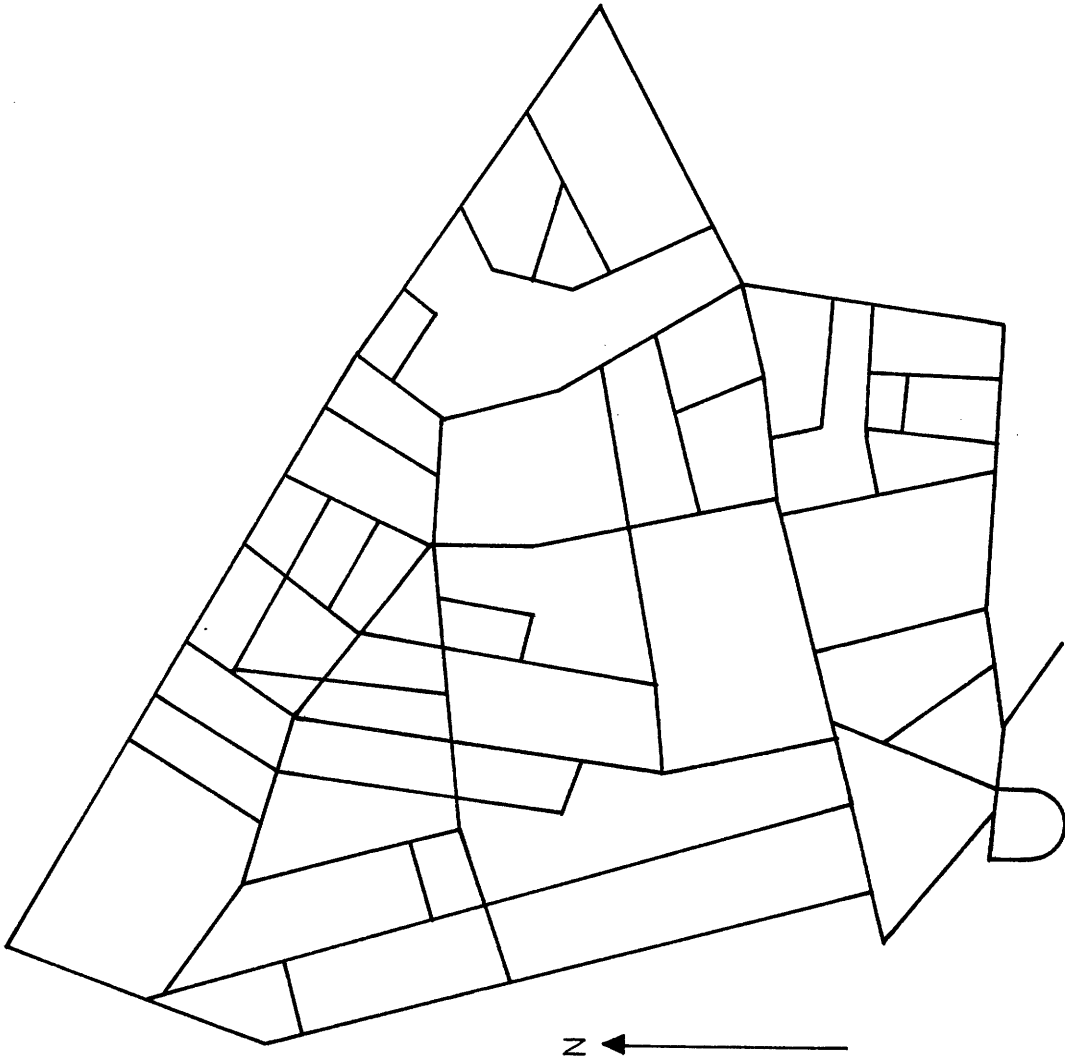


Figure 5-5. Map of Section II Including Boundary Edges.

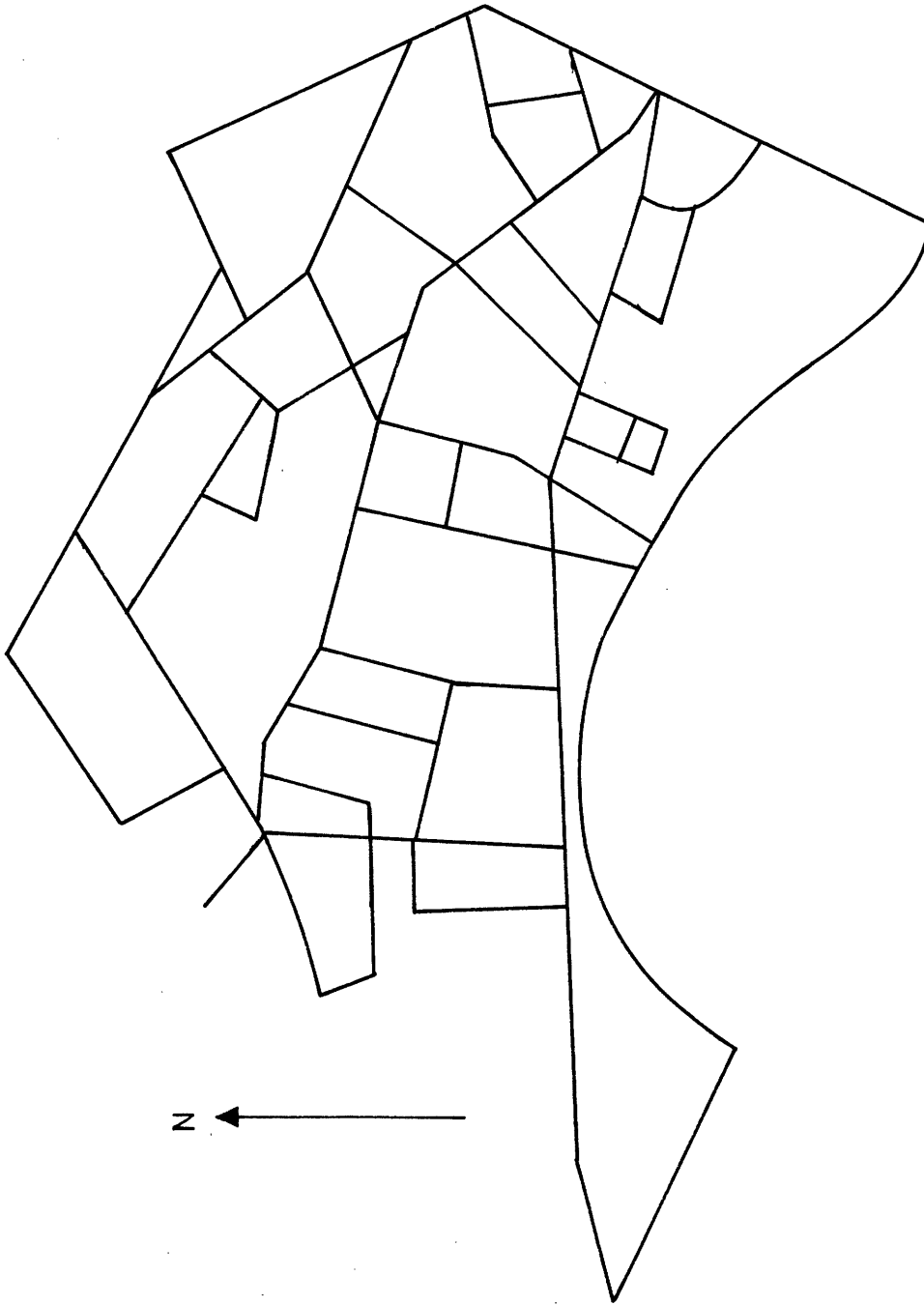


Figure 5-6. Map of Section III Including Boundary Edges.

considering that for 50 odd nodes there are $50!/2!48! = 1205$ possible pairings. While no explicit algorithm exists to determine initially which nodes to pair, experience indicates that a certain ability is developed for making good initial choices in pairing nodes. Fortunately, this ability is picked up quite quickly. In general, nodes tend to be paired to their nearest neighbors. Start off by pairing isolated pairs, and then work up to small groups. It is helpful to draw in the duplicated arcs using a bright color so they stand out. This makes it easy to spot negative cycles by looking for concentrations of the color used.

Experience indicates that once the first two nodes in a group are matched, the rest follow naturally. If unable to logically make this initial decision, simply choose arbitrarily. If the wrong choice is made, negative cycles quickly become apparent. When negative cycles occur, go back to where the last arbitrary decision was made and try again. Despite the large number of possible feasible pairings, few ever enter into contention. This is because it is highly unlikely that a node in one end of the network will be matched to one in the other end. In the situations where the wrong two nodes are matched, negative cycles turn up quite rapidly, and are quickly spotted if a bright color is used to draw duplicated edges. Unfortunately, the problem of spotting negative cycles, which turns out to be almost a pattern recognition problem, while quite simple for the human mind, is almost impossible to program on a digital computer with today's technology. Thus, the two hours required to solve this problem by hand could not be reduced to seconds, as one would imagine, by programming the algorithm using present techniques.

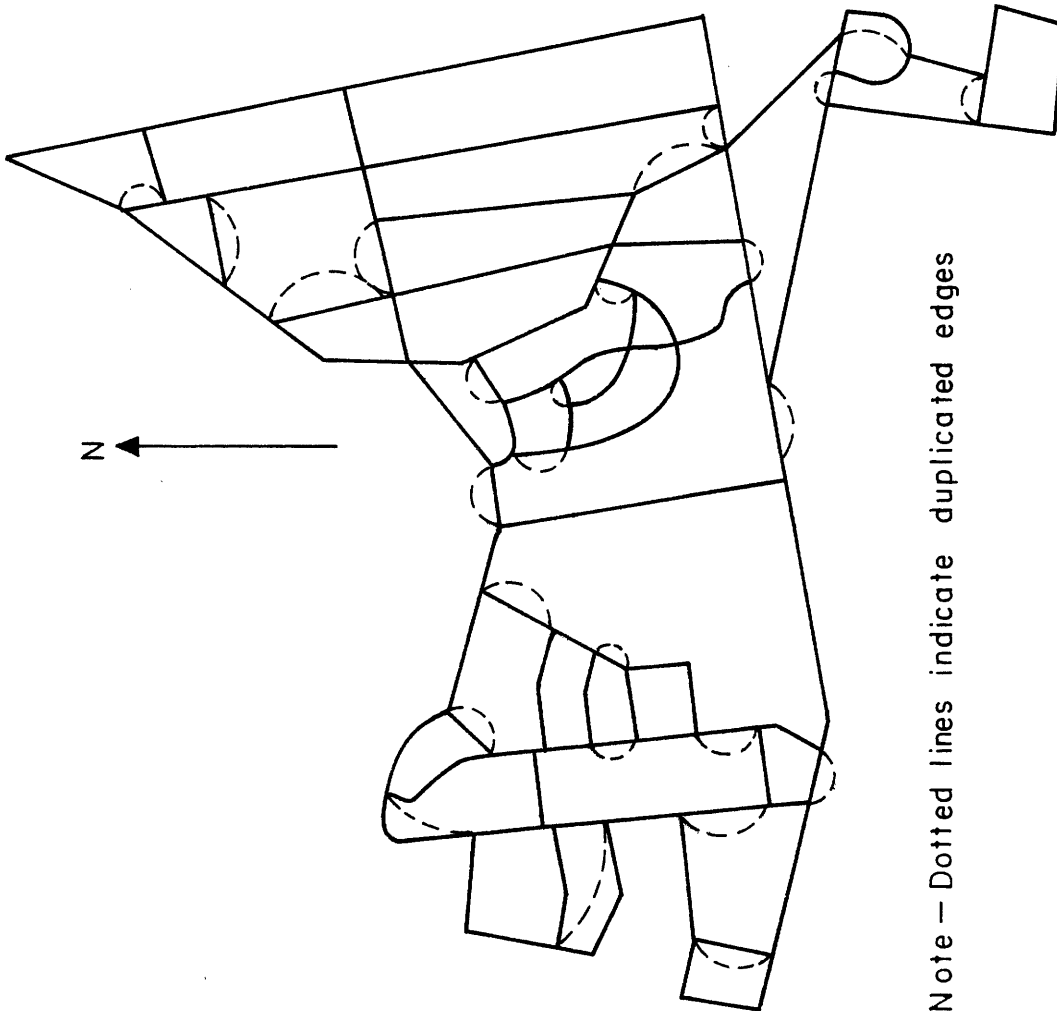
Unless advances are made in computer programming technology with respect to pattern recognition problems, this algorithm will continue to be feasibly executed only by hand. For a computer to determine the presence of negative cycles would require checking every existing cycle, which means millions of calculations for any large network.

Figures 5-7, 5-8, and 5-9 show optimal solutions to the Chinese postman problem for sections I, II, and III respectively. Note that only in rare cases is a shortest path of duplicated edges connecting two odd nodes in the optimal solution composed of more than a single edge joining the two odd nodes.

Figure 5-10 results from combining solutions to sections I, II, and III. It is encouraging that no negative cycles are present. This shows the possibility of partitioning a network so that when the optimal solutions for each section are combined, the optimal solution to the entire network results. Even when such an ideal partitioning is not produced, one can see from the previous trial and error solutions of the subproblems that finding negative cycles until the optimal solution is obtained is not that difficult a task.

3. Results

The computational results for this algorithm are encouraging. The sample problem with about 250 edges and 144 odd nodes took less than three hours to completely solve by hand, which was much less than had been anticipated. Much of this time was spent drawing in, and to some extent, erasing edges. The longest step by far was solving the Chinese



Note — Dotted lines indicate duplicated edges

Figure 5-7. Solution of Chinese Postman Problem for Section I.

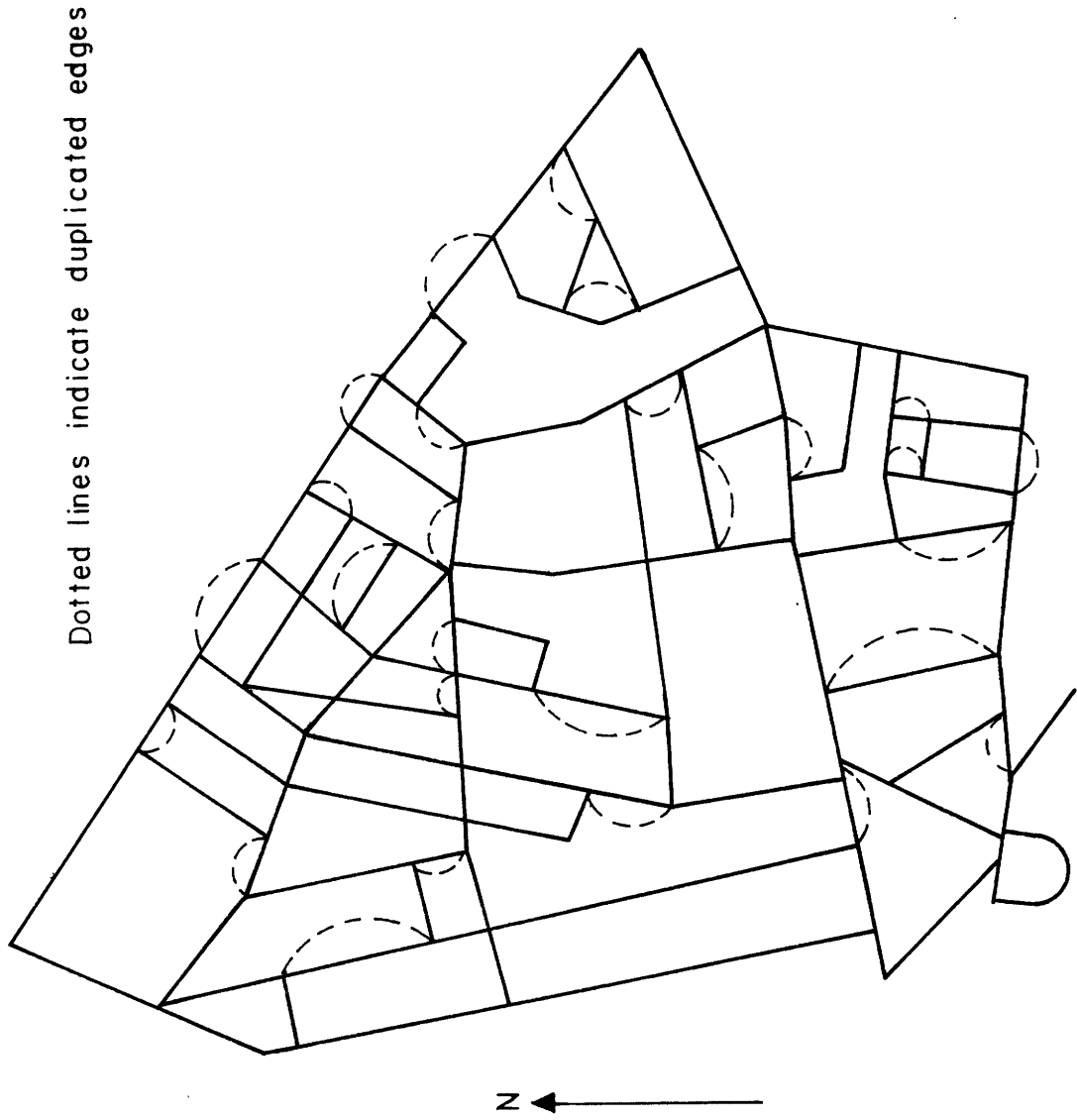


Figure 5-8. Solution of Chinese Postman Problem for Section II.

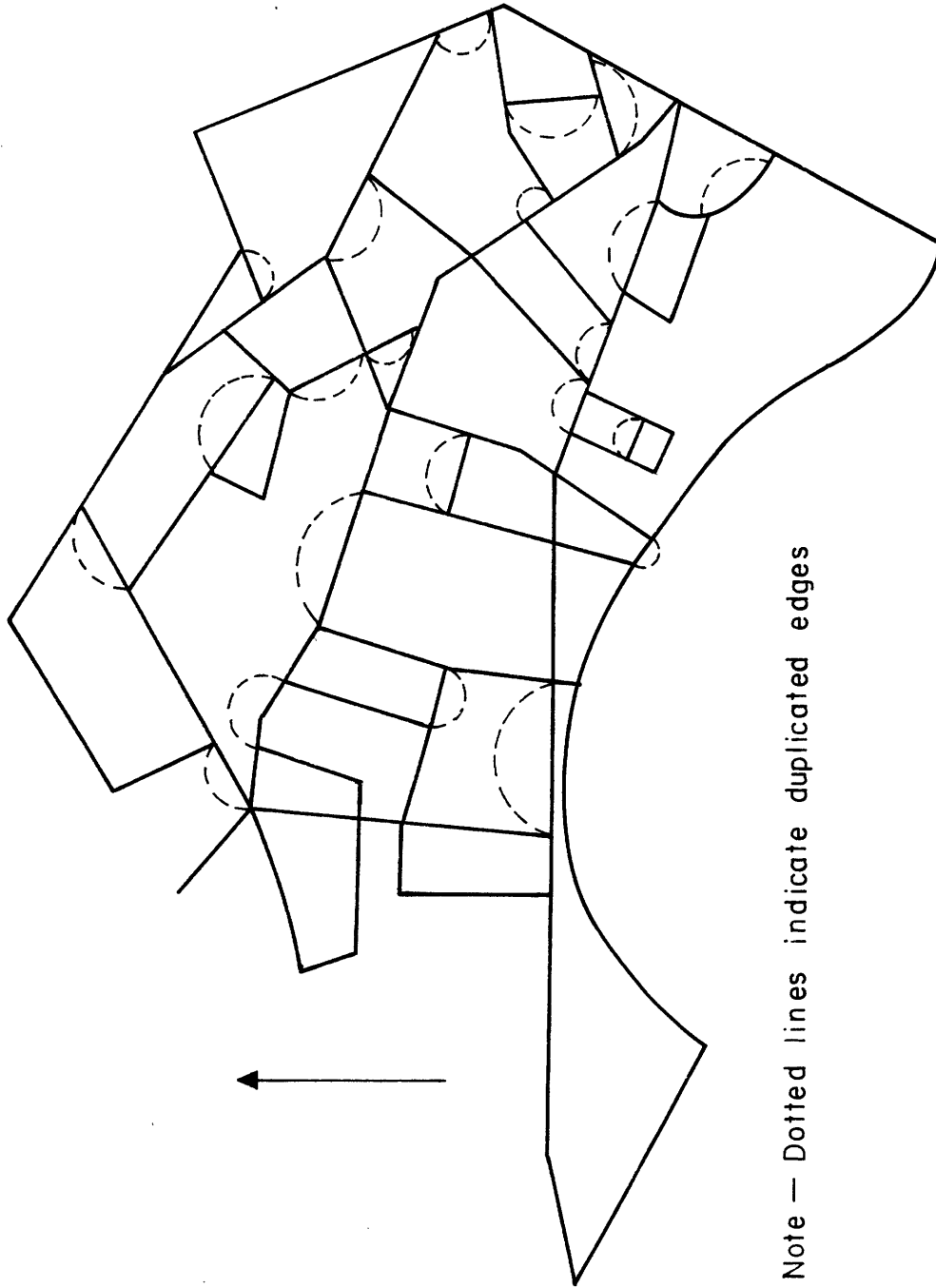
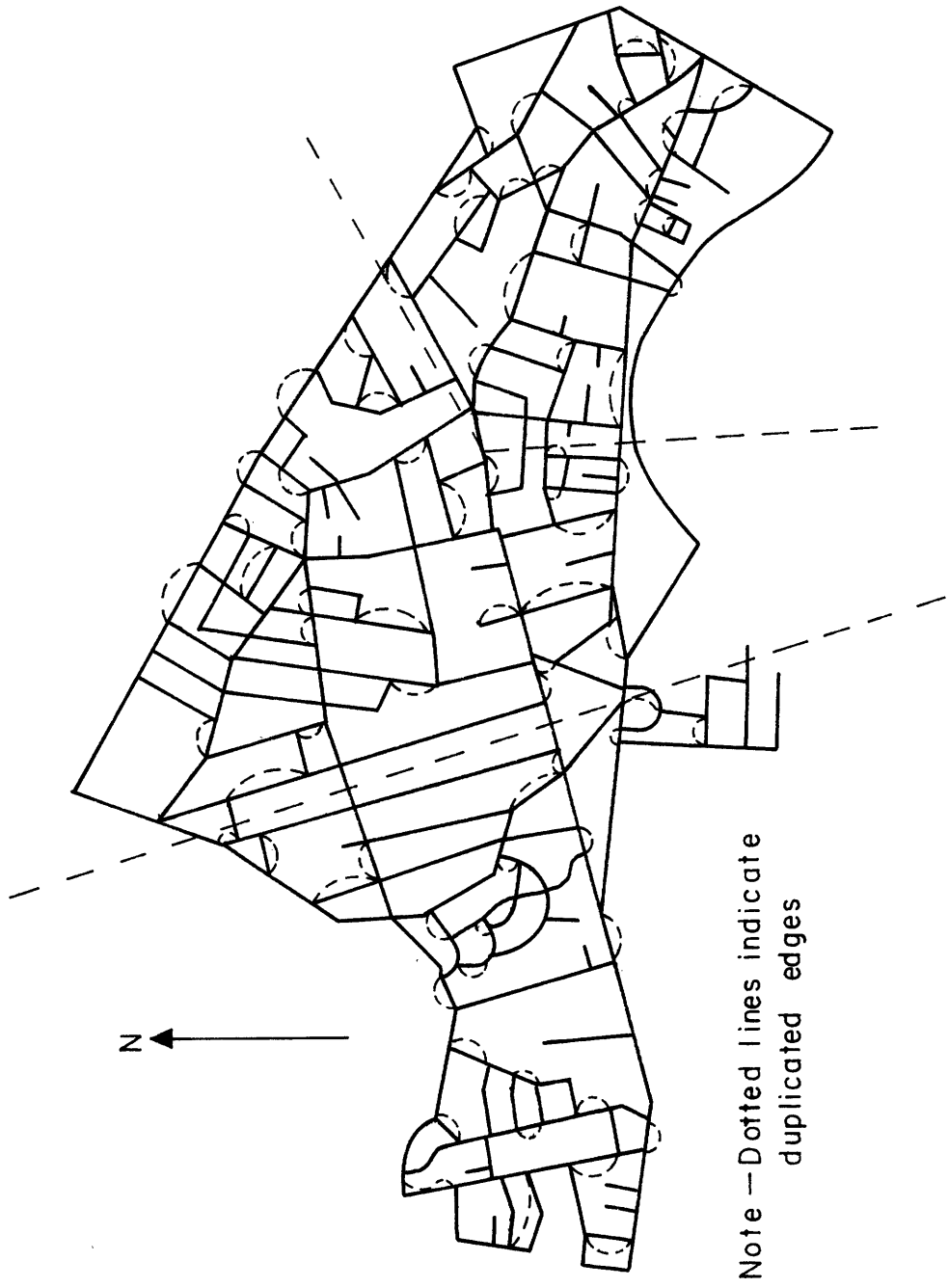


Figure 5-9. Solution of Chinese Postman Problem for Section III.



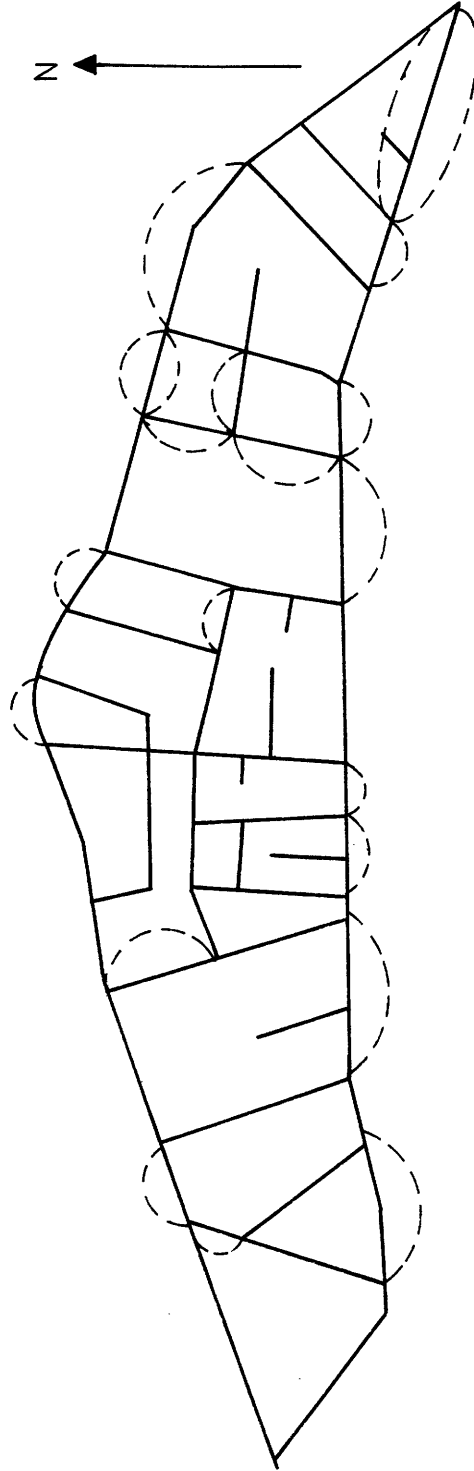
Note — Dotted lines indicate duplicated edges

Figure 5-10. Solution of Chinese Postman Problem for Entire Network.

postman problem for each section. However, as mentioned previously, one quickly develops a knack for the trial and error method, so that subsequent sections take much less time than earlier ones.

It is reasonable to assume that the solution presented in Figure 5-10 is optimal, or at least extremely close to optimal, since after an initial satisfactory solution was obtained in less than three hours, four subsequent half hour sessions were spent searching for negative cycles, but none were found. It is therefore the opinion of the author that such additional searching is unnecessary. Negative cycles should stand out immediately. Additional confidence may be placed in the solution, since of the 72 shortest paths necessary to match the 144 odd nodes, only seven require two edges, and none require more than two. Moreover, the great majority of nodes are matched to their nearest odd node, which is reassuring.

Of course, the true test of the value of a new approach to a problem is to determine the savings it produces from the existing way the problem is solved. Unfortunately, the city of Cambridge does not maintain set routes for the collection vehicles to follow. Therefore, to obtain data on which to base comparisons, the author spent two days riding along with a trash truck and tracing out the route covered. Figure 5-11 presents the resulting route, which covers only about one-fifth of the section of the city used for the sample problem. Using the solution from Figure 5-10, this area with 4.73 miles of street requires duplicating 1.01 miles to cover it. However, from Figure 5-11, 1.87 miles are presently duplicated to cover this section. The solution represents a saving of 46 percent of



Note - Dotted lines indicate duplicated shortest paths. Not all odd nodes are matched since two vehicles covered this area in five trips altogether

Figure 5-11. Map of Area Used as Basis for Comparison.

unnecessarily duplicated streets, and 13 percent of total distance travelled. Since the route covered was picked at random, there is no reason to doubt that this type of savings would not occur for the city as a whole if the decomposition algorithm were applied to determine routes for all the collection vehicles. With collection and disposal costs amounting to \$1.5 million per year in Cambridge, any significant increase in efficiency results in considerable cash savings. Besides the savings in reduced wear and tear on vehicles resulting from reducing mileage, a single crew can cover more streets, so one or two less crews are needed to cover the city.

CHAPTER VI. CONCLUSIONS AND RECOMMENDATIONS

The problem of finding the shortest route through a network so that every edge is covered at least once has interested men for centuries. In modern times, with variations of this problem such as trash collection and snow plowing facing urban planners, it becomes increasingly important that efficient means to solve these problems be found. Unfortunately, while Euler first attacked the problem in 1736, and while several recent researchers have developed algorithms which in theory are capable of finding the optimal solution to the theoretical problem, no algorithm yet appears in the literature capable of efficiently handling the size problems currently faced. Moreover, no author even mentions important variations such as capacity constraints and priorities of main arteries in snow plowing.

The purpose of this thesis as stated in the introduction was to study real world problems and to develop techniques to solve them. In accomplishing this purpose it is felt that this thesis has been at least moderately successful. The decomposition algorithm presented, although to some extent heuristic and requiring a bit of art to execute, is still capable of generating at least a good if not optimal solution to the problem quickly and efficiently. To guarantee optimality of the final solution, it is possible to apply a shortest path algorithm such as the one proposed by Dantzig, Blattner, and Rao^[3] to determine whether any negative cycles exist in the network. Remember that if no negative cycles exist, the solution must be optimal. It is the author's

opinion, though, that such additional checks are not worth the added computational effort. While currently the trend is to computerize everything, it was felt that this was not advisable for the decomposition algorithm for two reasons. First of all, computer programming technology has not yet reached the stage where it is capable of efficiently handling the pattern recognition problems presented by the algorithm. The only way a computer could check for negative cycles would be by exhaustive enumeration, which is unthinkable. Secondly, the time required to change the data base from its existing map form into an array which could be fed into a computer using normal I/O devices would probably be more than the three hours it took by hand to completely solve the problem presented in Chapter V.

As for developing techniques to handle practical variations of the theoretical problem, this thesis was somewhat less successful. The m-postmen Chinese postman problem algorithm for handling the capacity constraint, while certainly a step in the right direction, has several drawbacks. Moreover, no explicit algorithm was even developed to handle the problem of street priorities in snow plowing. The important thing, though, is that these additional constraints have been formally discussed so that the problems which need to be solved are at least clear.

Certainly, much research needs to be done. Edmonds' as yet unpublished work would seem to hold promise for the future. Another area which should be looked into is network flow algorithms. With the large amount of structure present in the Chinese postman problem, it may be possible by some fancy manipulation of the problem to apply the out-of-

kilter algorithm which may be obtained from IBM. One possibility is to divide the n odd nodes into two sets with $n/2$ nodes in each. Place unit inputs on each node in one set, unit outputs on each node in the other set, and force a flow of $n/2$ units through the network to determine the minimum pairing. Since there would be $n!/(\frac{n}{2})!(\frac{n}{2})!$ such minimum pairings, checking each one to determine the minimum of the minimums is unfeasible, but perhaps this scheme could serve as the basis for a heuristic algorithm, or perhaps even a branch and bound algorithm. This approach is similar to the one by Murty and Edmonds, since a minimum pairing is sought. However, it differs because at each iteration a feasible solution is available since a minimum pairing of two different mutually exclusive, collectively exhaustive subsets of nodes is found. The two subsets whose minimum pairing is the minimum of all those obtained, is the optimal solution to the problem.

The problems in the area of vehicle routing are many. Much work has been done in the field, but as in most areas of engineering endeavor, a great deal remains to be done.

APPENDIX I. CONSTRUCTION OF AN EULER CHAIN

An Euler chain is a path through a network which covers every edge exactly once. The following theorem is an existence theorem for an Euler chain in a network.

Theorem

A graph G possesses an Euler chain if and only if it is connected, and the number of nodes with uneven degree is 0 or 2, where the degree of a node refers to the number of edges incident to it.

The proof of this theorem is not difficult. The necessity of the condition is obvious if one considers that if a graph has an Euler chain u , then it is connected, and only the two end points of u , if they are distinct, can be odd. Sufficiency of the theorem is proven by induction. Assume the statement holds true for a graph with fewer than m edges, and then show that an Euler chain exists whose end points are the two odd nodes, a and b , for a graph G with m edges. (An Euler tour or cycle exists if there are no odd nodes.) The chain will be defined by starting at one of the odd nodes, a , and covering edges in such a way that the same edge is not covered twice. To ensure this, erase an edge once it is covered. Because every node except b is even, it will always be possible to leave a node once it is entered, and its parity remains unchanged because two edges are always erased at a time. Once having entered a node with no remaining edges to leave by, it must necessarily be b . It is possible that not every edge in G was covered. However, for the graph

G' of remaining edges, all of its nodes must be even and their number less than m , so by assumption each component of G' must contain an Euler cycle. Because the original graph G was connected, each of the Euler cycles in G' must be connected to the original chain from a to b in G . By considering the union of all these chains and cycles, an Euler chain can be defined from a to b in G .

The above proof is very convenient in that it specifies an efficient algorithm to construct an Euler chain/cycle from a graph with two/zero odd nodes, call them a and b .

Algorithm to Construct an Euler Tour

- (1) Start at odd node a (or at any node if no odd nodes exist) and erase each edge as it is covered.
- (2) Do not cover an edge that is an isthmus. An isthmus is an edge whose erasure would divide the graph into two non-empty components.

The above rules can always be followed because there is always an edge leaving from the current node X ($X \neq b$) which is not an isthmus. If all the edges incident to X were isthmuses, of which there must be at least two, then they would lead to two disjoint components, each of which contain an odd vertex, which is a contradiction.

As an example of this algorithm, consider Figure I-1. The task is to draw an Euler chain on this graph, starting at node a . Because every node except a and b is even, this must be possible. One optimal solution (see Figure I-2), given in the order in which the nodes are visited, is $\{a, c, f, e, c, d, b, e, d, a, b\}$. Of course, many other optimal solutions, such as $\{a, d, c, e, d, b, e, f, c, a, b\}$ and $\{a, d, e, c, f, e, b, c, a, b\}$ exist. The fact

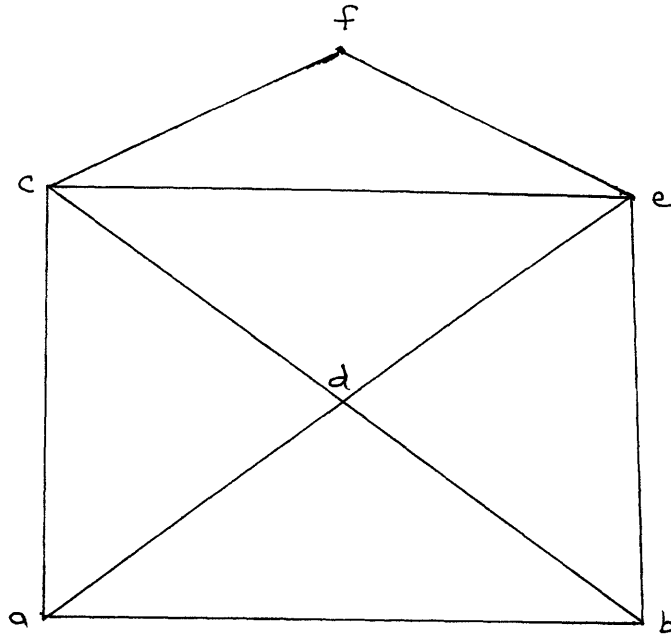
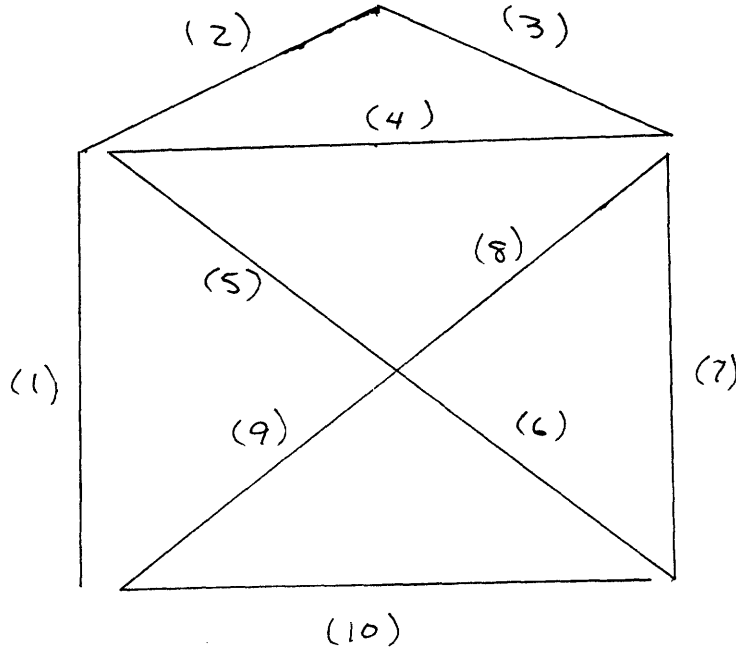


Figure I-1. Example of a Graph with Two Odd Nodes.



Note - Numbers in parenthesis refer to order in which edges are drawn.

Figure I-2. Euler Tour for Network in Figure I-1.

that alternate optima exist in these problems may be helpful when attempting to solve more constrained problems such as those with capacity constraints, and those with one way streets.

APPENDIX II. GLOVER'S ALGORITHM

Before formally presenting Glover's algorithm and working a short example, certain remarks and theorems will be given, and several additional terms such as pseudo complement, real base of a graph, and pseudo translate will be defined. The reader already familiar with this area may pass over this introductory material and go directly to the algorithm itself.

Background Material

For a graph P consisting entirely of pseudo edges, define its pseudo complement, \bar{P} , as the graph of ordinary edges corresponding to the pseudo edges of P . In mathematical terms,

$$\bar{P} = \sum_{\substack{S \\ pq \in P}} S_{pq} \quad (7)$$

where the sum counts elements of P according to their multiplicity (if S_{pq} appears k times in P , then S_{pq} appears K times in the sum).

Conversely, given a graph H consisting entirely of ordinary edges, a graph P consisting entirely of pseudo edges is said to be the pseudo complement of H , $\bar{H} = P$, if the pseudo complement of P as defined above equals H ($\bar{P} = H$). From this set of definitions it is important to note that while the pseudo complement of a graph of pseudo edges is unique, this generally will not be true for a graph H of ordinary edges. The reason is that there usually exists more than a single unique path between

every pair of nodes, so arbitrarily choosing paths corresponding to every edge in H will result in different pseudo graphs P_1, P_2, \dots, P_m , all with pseudo complement H . Thus the pseudo complement of a graph H of ordinary edges is not usually unique.

The notion of a pseudo complement can be extended to any arbitrary graph X . Define $F(X)$ to be the graph of pseudo edges in X , and $T(X)$ to be the graph of ordinary edges in X , always being careful to conserve multiplicity. The following relations are obvious:

$$F(X) + T(X) = X \quad (8)$$

$$F(X) \cap T(X) = \phi \quad (\text{null set}) \quad (9)$$

Now define a pseudo complement \bar{X} of X as

$$\bar{X} = \overline{F(X)} + \overline{T(X)} . \quad (10)$$

Note that since the complement of a real graph need not be unique, there may be more than one \bar{X} if $T(X)$ is not empty. Two important characteristics of a pseudo complement are

$$X + \bar{X} \text{ is even,} \quad (11)$$

$$\ell(X + \bar{X}) = 0 \quad (12)$$

Based on these definitions, Glover proves the following theorem as an aid in developing an algorithm to solve the Chinese postman problem.

Theorem

Let H be any subset of edges of graph G such that $G+H$ is even. Let \bar{H} be any pseudo complement of H . Then H is an optimal solution for the Chinese postman problem if and only if $G+\bar{H}$ has no simple cycles of negative length.

It is quite apparent that this result is equivalent to the one by Mei-Ko presented earlier. Since repeated arcs are negative and negative cycles must be avoided, this amounts to saying that duplicated arcs may not amount to more than half the length of any cycle. Obviously, this new statement of the theorem alone is no better in proposing an algorithm for the Chinese postman problem. However, by incorporating it along with several new functions defined below, a much more efficient algorithm can be developed.

Define the real base X^* of a graph X to be the graph

$$X^* = T(X) + \overline{F(X)}. \quad (13)$$

In other words, obtain X^* from X by replacing each pseudo edge of X with its corresponding simple path of ordinary edges. If X has no pseudo edges, then $X^* = X$. The pseudo translate $(X: P)$ of a graph X with respect to a simple path $P \subset X$ is

$$(X: P) = X - F(P) + \overline{T(P)}. \quad (14)$$

To obtain $(X: P)$ from X , remove the pseudo edges of P and then add pseudo edges corresponding to the ordinary edges of P .

The five remarks below follow immediately from the above definitions, where P is any simple path from node p to node q ($p \neq q$) in X .

Remarks

1. $T(X: P) = T(X)$
2. $(X: P)^* = X^* - \overline{F(P)} + T(P)$
3. $\ell((X: P)^*) = \ell(X^*) + \ell(P)$
4. nodes p and q are even in $(X: P)^*$ if and only if they are odd in X^*
5. node r , $r \neq p, q$ is even in $(X: P)^*$ if and only if it is even in X^* .

With this new terminology explained, Glover's algorithm can now be presented.

Glover's Algorithm

1. To start, let $K=0$, $G_0=G$, and O_0 be the set of odd nodes of G .
2. Select any two nodes $p_k, q_k \in O_k$, and identify a shortest elementary path P_k between p_k and q_k in G_k .
3. Let $G_{k+1} = (G_k: P_k)$ and $O_{k+1} = O_k - \{p_k, q_k\}$.
4. Increment k by 1. If $k=m$ (where $2m$ is the original number of odd nodes in G), then O_m is empty, and the graph $\overline{F(G_m)}$ is the solution we seek. The optimal solution to the Chinese Postman's Problem is obtained by determining an Euler tour on G_m^* . Otherwise, if $k < m$, return to instruction 2.

Glover^[10] presents a rigorous as well as quite elegant proof of his algorithm, and the interested reader is referred directly to his original

work. However, to give a little more insight into the workings of his algorithm, the following intuitive justification, first suggested by Renard^[22], is presented.

First it will be demonstrated how the algorithm generates the correct solution under a particular set of circumstances, and then it will be shown what happens under more general circumstances. Recalling Mei-Ko's work, the optimal solution for a graph with $2m$ odd nodes will require duplicating edges to create m paths connecting m pairs of these nodes. No edge will be duplicated more than once in the optimal solution. Consider the case where by chance the two nodes chosen from O_0 , which are joined by path P_0 in G_0 , are to be joined by a path of duplicated edges in the optimal solution. If the same type of chance selection occurs in O_1 , then for the path P_1 in $G_1 = (G_0 : P_0)$ which connects the two new nodes, $F(P_1)$ will not contain any edges since P_1 and P_2 are disjoint in the optimal solution. Hence, for $G_2 = (G_1 : P_1)$,

$$G_2 = G_0 + \overline{T(P_0)} + \overline{T(P_1)} \quad (15)$$

where $T(P_1)$ and $T(P_2)$ are determined on G_0 . If two nodes in O_k are always picked which are joined in the optimal solution, then after the m^{th} iteration,

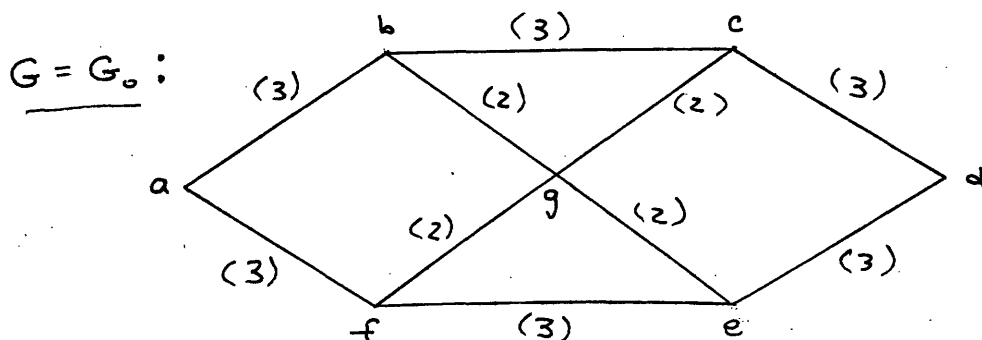
$$G_m = G_0 + \overline{T(P_1)} + \overline{T(P_2)} + \dots + \overline{T(P_m)} \quad (16)$$

It is readily apparent that G_m^* is the optimal graph on which to determine an Euler tour.

In the more typical case, when two nodes are chosen which are not connected in the optimal solution, it must be determined how the algorithm functions. In such a case, when taking the pseudo translate of the graph with respect to the path P_1 , pseudo edges are added corresponding to $\overline{T(P_1)}$. By adding these edges with negative length, an added attraction has been added for an upcoming shortest path, P_k , to take these edges. If one such path, P_r , does take one of these pseudo edges associated with P_1 , then obviously P_1 and P_r are not an optimal pairing of the four nodes comprising their end points since they share an edge. However, this is reflected in $G_{r+1} = (G_r : P_r)$ since the pseudo edges from previous paths in G_r which are associated with P_r , those edges shared by more than one path, are removed in forming G_{r+1} . This process continues each time a new pair of nodes is added.

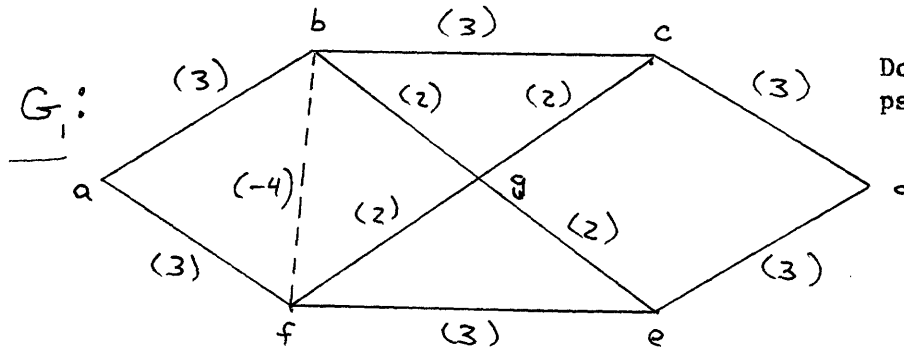
Example

At this point it would probably be beneficial to apply this algorithm to an example problem. Consider the following graph G .



The numbers in parentheses indicate the cost associated with each edge.

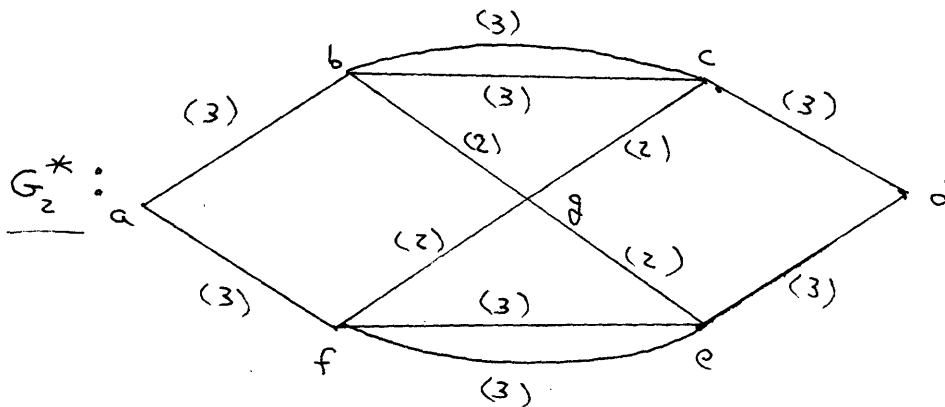
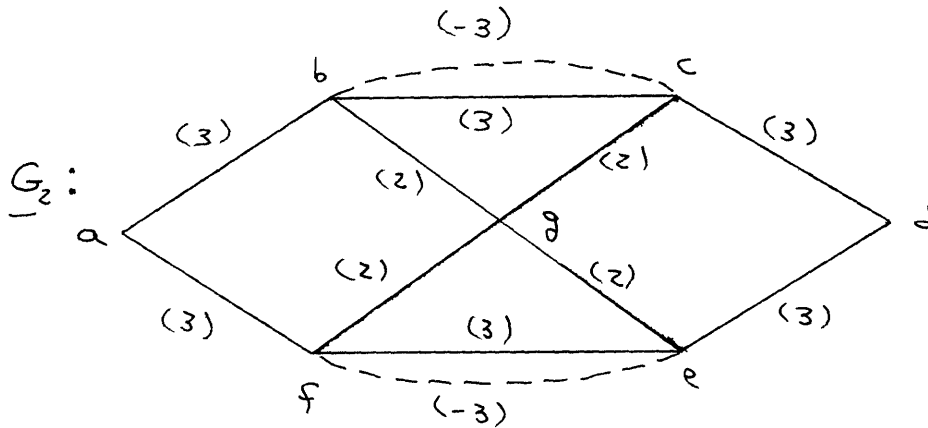
1. $O_0 = \{b, f, c, e\}$, so arbitrarily let $p_0 = b$ and $g_0 = f$.
2. $P_0 = (b, g, f)$ by inspection. $F(P_0) = \phi$.
3. $G_1 = (G_0 : P_0) = G_0 + \overline{(b, g) + (g, f)}$, and $O_1 = \{c, e\}$, so $p_1 = c$ and $q_1 = e$.



Dotted lines indicate pseudo edges.

4. $P_1 = (c,b,f,e)$ by inspection. $F(P_1) = (b,f)$.

5. $G_2 = (G_1 : P_1) = G_1 - (b,f) + \overline{(b,c)} + \overline{(f,e)}$, and $O_2 = \phi$, so G_2^* is the graph we seek.



Thus edges (b,c) and (f,e) must be duplicated to obtain the optimal solution.

Of course, this result could have been obtained very simply by inspection, but it is reassuring to see how easily it is obtained using Glover's algorithm. Note at this point that although for this simple problem it was possible to determine the shortest path between two nodes by inspection, in life size problems this would not be possible. In general it would be necessary to apply one of the well known shortest path algorithms, or else use the algorithm Glover develops in [10] especially for his algorithm for the Chinese postman problem.

APPENDIX III. MURTY'S MATRIX REDUCTION METHOD AND BRANCHING RULES

Murty's matrix reduction method has two parts, which are repeated iteratively if necessary. First start with matrix C_0 by finding its optimal assignment using the Hungarian method developed by Kuhn^[14,15]. The following simplified version of the algorithm is adapted from Hillier and Lieberman^[11]:

- (1) Subtract the minimum element in each row in the cost matrix from every element in that row. Do the same for each column.
- (2) Examine the rows and columns successively. For each row/column with exactly one zero element, mark that element reserved and eliminate other zero elements in its column/row from further consideration. Continue for rows and columns without reserved positions until all zero element positions are either reserved or eliminated. If the reserved positions specify a complete assignment, it is an optimal solution. Otherwise, go to step (3).
- (3) Draw a minimum number of lines to cover all zero elements as follows:
 - (a) Mark all rows that do not have assignments.
 - (b) Mark all columns which have zeros in marked rows.
 - (c) Mark all rows that have assignments in marked columns.
 - (d) Repeat steps (b) and (c) until no more rows or columns can be marked.
 - (e) Draw a line through each unmarked row and through each marked column.

- (4) Subtract the minimum element not covered by a line from all such uncovered elements. Then add it to each element that lies at the intersection of two lines. Go to step (2).

The final matrix obtained from C_1 with the zeros specifying a complete assignment will be called C_{i+1} , where the initial cost matrix will be assumed to be C_0 .

The second part of the reduction method is to repeat the first part if C_1 is not symmetric, with $C'_1 = \frac{C_1 + C_1^T}{2}$ in place of C_0 . Repeat this process of finding the optimal assignment with respect to transformed matrices as many times as necessary until the final transformed matrix, C_R , with at least one zero in each row and column, is symmetric. The sum of the costs of all the optimal assignments as they are obtained in successive steps is the reduction of the matrix C .

Fortunately, Murty proves that the process needs to be repeated twice at most, and the cost of the first optimal assignment is actually the reduction of matrix C .

Before presenting Murty's branching rules, two additional definitions must be given. The remaining cost matrix at node N , where N is defined according to the previous definition in Chapter III, section 5, is obtained by crossing out rows and columns $i_1, \dots, i_r, j_1, \dots, j_r$ of C , and replacing the cost elements in the cells $(m_1, p_1)(p_1, m_1) \dots (m_s, p_s)(p_s, m_s)$ by infinity (or an appropriately large number). The reduced matrix obtained from this matrix is known as the reduced remaining cost matrix, and is written $C_{N,R}$.

Finally, the evaluation of an admissible unspecified cell (i,j) at a node N is defined to be $\theta_N(i,j)$ where

$\theta_N(i,i)$ = Sum of the minimal elements in row i and column i of $C_{N,R}$ after excluding the element in (i,i) ;

$\theta_N(i,j)$ = Sum of the minimal elements in rows i and j and columns i and j in $C_{N,R}$ after excluding the (i,j) th and (j,i) th elements, but only if these minima occur at distinct places.

If these minima are not all distinct, but if two coincide at one or both of the diagonal cells (i,i) and (j,j) , simply sum the diagonal cell(s) where the minima coincide, plus the minimal elements in the row and column (if any) whose minima are distinct. In other words, do not count any cell in the sum more than once, even if two minima are coincident there.

Murty's branching rules are as follows:

- (i) Find the minimal terminal node N with the least cardinality (the one with the maximum number of cells specified to be contained in it).
- (ii) Find (i,j) , an admissible unspecified cell at node N such that
 - (a) (i,j) is a zero cell in $C_{N,R}$, and
 - (b) $\theta_N(i,j)$ is maximum among the cells satisfying (a).
- (iii) Branch from N with respect to (i,j) and find the lower bounds on both the branches using

$LB(N_1) = LB(N) +$ reduction of $C_{N,R}$ without rows i,j or columns i,j

$LB(N_2) = LB(N) +$ reduction of $C_{N,R}$ with cells (i,j) and (j,i) replaced
by infinity (or a very large number).

(iv) Go to stage $m+1$.

These computations are repeated until the optimality criterion is satisfied.

BIBLIOGRAPHY

1. Berge, Claude, The Theory of Graphs and Its Applications, John Wiley and Sons, Inc., New York, 1966.
2. Busacker, R. G., and Saaty, T. L., Finite Graphs and Networks; an Introduction with Applications, McGraw-Hill, New York, 1965.
3. Dantzig, G. B., Blattner, W. O., and Rao, A. O., "All Shortest Routes from a Fixed Origin in a Graph," Operations Research House, Stanford University, Technical Report 66-2, November, 1966.
4. Dreyfus, Stuart E., "An Appraisal of Some Shortest-Path Algorithms," Operations Research, 17, 395-412, 1969.
5. Edmonds, Jack, "Optimum Branchings," Journal of Research National Bureau of Standards, 71B, 223-240, 1967.
6. Edmonds, Jack, "Paths, Trees, and Flowers," The Canadian Journal of Mathematics, 17, 449-467, 1965.
7. Edmonds, Jack, "Maximum Matching and a Polyhedron with (0,1) Vertices," Journal of Research National Bureau of Standards, 69B, 1965.
8. Edmonds, Jack, "An Introduction to Matching," Mimeographed Lectures, Ann Arbor, Michigan, 1967.
9. Floyd, R. W., "Algorithm 97, Shortest Path," Comm. ACM, 5, 345, 1962.
10. Glover, Fred, "Finding an Optimal Edge-Covering Tour of a Connected Graph," ORC 67-13, Operations Research Center, University of California, Berkeley, 1967.
11. Hillier, F. S., and Lieberman, G. J., Introduction to Operations Research, Holden-Day, Inc., San Francisco, 1969.
12. Hu, T. C., "A Decomposition Algorithm for Shortest Paths in a Network," Operations Research, 16, 91-102, 1968.
13. Johnson, Ellis L., "Existence of Euler Tours in Bidirected Graphs," IBM Research, RC 2753, Yorktown Heights, 1970.
14. Kuhn, H. W., "The Hungarian Method for Solving the Assignment Problem," Naval Research Logistics Quarterly, 2, 83-97, 1955.
15. Kuhn, H. W., "Variants of the Hungarian Method for Assignment Problems," Naval Research Logistics Quarterly, 3, 253-281, 1956.

16. Land, A. H., and Stairs, S. W., "The Extension of the Cascade Algorithm to Large Graphs," Management Science - A, 14, 29-36, 1967.
17. Ludwig, H. F., and Black, R. J., "Report on the Solid Waste Problem," Journal of the Sanitary Engineering Division, American Society of Civil Engineers, 94, SA2, April 1968.
18. Marks, D. H., and Liebman, J. C., "Mathematical Analysis of Solid Waste Collection," Department of Geography and Environmental Engineering, The Johns Hopkins University, 1970.
19. Mei-Ko, Kwan, "Graphic Programming Using Odd or Even Points," Chinese Mathematics, 1, 273-277, 1962.
20. Mills, G., "A Decomposition Algorithm for the Shortest Route Problem," Operations Research, 14, 279-291, 1966.
21. Murty, K. G., "The Symmetric Assignment Problem," ORC 67-12, Operations Research Center, University of California, Berkeley, 1967.
22. Renard, O., "The Chinese Postman Problem," Term Paper for course Introduction to Optimization Techniques, Massachusetts Institute of Technology, 1969.
23. Yen, J. Y., "Matrix Algorithm for Solving All Shortest Routes from a Fixed Origin in the General Networks," presented at the Second International Conference on Computing Methods in Optimization Problems, San Remo, Italy, 1968.