A HEURISTIC SOLUTION PROCEDURE FOR THE CONSTRAINED NODE ROUTING PROBLEM AND AN IMPLEMENTATION TO HAZARDOUS SOLID WASTE COLLECTION

by

ALİ TÜMER AKYÜZ

B.S. in M.E. İTÜ, 1986

Submitted to the Institute for Graduate Studies in Science and Engineering in Partial fulfillment of the requirements for the degree of Master of Science in Industrial Engineering

ACKNOWLEDGEMENS

I would like to express my gratitude to my thesis supervisor, Prof. Dr. İlhan Or, for his support, encouragement, understanding, patience and tolerance throughout all phases of this study. He never deprieved me of his invaluable assistance and pointing critics which determined the basic frame of my study. Without his motivating endeavor, this study would not be as complete as expected.

I would also like to thank Prof. Dr. Gündüz Ulusoy, our Head of Department, for his motivating advices and keen interest in my study. My responsibilty against him urged me to do my best for this thesis.

I would also sincerely wish to thank Doç. Dr. David Pinhas and Doç. Dr. Attila Akkoyunlu for their valuable comments, suggestions and for serving on my thesis committe.

My special thanks are to my friend H. Mehmet Alpatli for his endless support.

ABSTRACT

A HEURISTIC SOLUTION PROCEDURE FOR THE CONSTRAINED NODE ROUTING PROBLEM AND AN IMPLEMENTATION TO HAZARDOUS SOLID WASTE COLLECTION

Constrained Node Routing is a well known Vehicle Routing Problem which has many real-life applications. In this study, first the general VRP is introduced together with solution procedures and its application to Solid Waste Collection is analysed. Then, CAVR (Computer Assisted Vehicle Routing), a new heuristic solution procedure used for solving the single depot-constrained node routing problems is presented. Its graphic displays and user friendly properties are also discussed. Thirdly, the developed procedure is tested on numerous literature problems so that an eveluation and comparison in terms of solution capabilities can be done. Finally, the suggested procedure is implemented to design routes for collecting the hazardous solid wastes of the hospitals in Istanbul.

ÖZET

KISITLANDIRILMIŞ NOKTA DOLAŞIM PROBLEMİ İÇİN BİR HÖRİSTİK ÇÖZÜM PROSEDÜRÜ VE TEHLİKELİ KATI ATIK TOPLANMASINA YÖNELİK BİR UYGULANIŞI.

Kısıtlandırılmış nokta dolaşım problemi geniş uygulama alanları olan, tanınmış bir Araç Yönlendirme Problemidir. Bu çalışmada, çözüm prosedürleri ile birlikte genel VRP tanıtılmış ve onun Katı Atık Toplanmasına uygulanışı çözümlenmiştir. Daha sonra, bir depolu-kısıtlandırılmış nokta dolaşım probleminin çözümünde kullanılan yeni bir höristik prosedür, CAVR (Bilgisayar Destekli Araç Yönlendirme), sunulmuştur. Onun grafik gösterimleri ve kullanıcıya kolaylık sa layan özellikleri de tartışılmıştır. Üçüncü olarak, çözüm kabiliyetleri açısından bir de erlendirme ve karşılaştırma yapılabilmesi için geliştirilen bu prosedür çok sayıda literatür problemi üzerinde denenmiştir. Son olarak, önerilen prosedür İstanbul'daki hastanelerin tehlikeli katı atıklarının toplanması için güzergahlar düzenlenmesinde uygulanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION AND BACKGROUND

Recognizing that our world is finite and that the continued pollution of our environment will, if uncontrolled, be difficult to reverse and correct in the future, the subject of solid waste management is both timely and important. The overall objective of solid waste management is to minimize the adverse environmental effects caused by generation and disposal of solid wastes. To assess the management possibilities it is important to consider materials flow in society, types of raw materials used, reduction in solid waste quantities, reuse of materials, materials recovery, energy recovery and logistics of solid waste management. While the listed issues are of great importance and provide a perspective on the waste problem in general, the fact remains that the logistics management of municipal solid wastes is a complex and costly undertaking. Direct activities that must be considered and coordinated within this context include on-site storage, collection , transfer and transport of solid waste, placement of processing and disposal facilities.

Solid waste management today is made difficult and costly by increasing volumes of waste produced, by the need to control potentially serious environmental and health effects of disposal and by the lack of land in urban areas for disposal purposes, partly due to public opposition to proposed sites. Waste management, once strictly a local and private sector matter, now involves regional authorities and many public interest groups. Various legislative initiatives and procedures have been activated, within the last few years, in the leading industrial countries.

1.1. The Solid Waste Collection Problem

Solid waste logistics management can be divided into two major sub-systems. One is the solid waste collection system and the other is the regional management system. A solid waste collection system is concerned with the collection of wastes from sources, routing for trucks within the region, the frequency of collection, crew size, truck sizes, number of operating trucks, transportation of collected wastes to transfer stations. The regional management concerns itself with the selection of the number and locations of transfer stations , intermediate processing facilities or landfill sites, their capacities, capacity expansion strategies and routing of the wastes through the facilities to ultimate disposal on a macroscopic level.

It is true that solid waste collection is an important face of overall waste management and is interrelated to the decisions and issues within the regional management context. The solid waste collection problem is the problem investigated in this study within the following framework : Given the potential locations of intermediate transfer facilities and landfills, the locations and capacities of existing facilities and the quantities of wastes generated at the sources, find out which facilities should be built and how the wastes should be routed and disposed of so that the overall transportation costs of the system are minimized. The other facets of the overall problem will not be covered here. These include questions of how to optimality organize the labor force, the effect of differences in collection technology, and the optimal location of disposal facilities. The said problem is modelled as a Vehicle Routing Problem. Within this context, in this study, first the general VRP is introduced together with its application areas and solution procedures, and its application to solid waste collection and removal is analysed. Then a solution procedure based on the most popular heuristic approaches in the literature and enriched through graphic displays and user interaction is presented. Finally, the suggested procedure is implemented to determine a good routing design for the vehicles collecting the hazardous solid wastes from the hospitals in Istanbul metropolitan area.

1.2. Arc and Node Routing Problems

Routing of solid waste collection vehicles is a vehicle routing problem. In the literature there are various models for vehicle routing. The Vehicle Routing Problem (VRP), the Travelling Salesman Problem (TSP) and the Chineese Postman Problem (CPP) are some specific examples and will be discussed in the next section. There are several things to look for in determining the type of model to be used in developing routes for the vehicles. Generally we are given a map of a region which can be transformed into a network with a node set N and arc set A. The fist questions that arises is whether we are trying to perform the routing over the nodes or over the arcs in the networks. The first class of problems have been called discrete or node routing problems while the second class of the problems have been called continuous or arc routing problems [1].

If there is only one vehicle to route over nodes in a network then we have the Travelling Salesmen Problem (TSP) which is an NP hard problem but for which there are very fast and effective heuristic solution procedures exist ([2],[3],[4],[5]). If one vehicle is available to route over the arcs of a network then we have a Chinese Postman Problem (CPP). When all arcs in the network are either directed or undirected and in the case of one vehicle then exact procedures are available for solving this problem ([6],[7],[8],[9]). In the case when the network contains both directed and undirected arcs the problem is NP hard and there are no exact optimizing procedures, however, there are some effective heuristic solution procedures [8].
On the other hand, an arc routing problem can be formulated as a Standard Vehicle Node Routing Problem (SVNRP) or shortly

Vehicle Routing Problem (VRP). This allows us to transform arc routing problems into node routing ones. This is of interest since node routing problems have received much more attention than arc routing problems in the research literature. The transformation might be considered as a means for linking the two groups of node and arc routing problems so that theoretical results on the former class can be extended to the latter class. However, claims are made as to the computational utility of this transformation. Future computational work has to determine whether the strategy of transforming an arc routing problem and solving it with a node routing algorithm is computationally effective or not [10].

Both the node and arc routing procedures become more complicated when it is necessary to route more than one vehicle over the network. In such a case, one have two choices : First, one can cluster the networks into subgraphs and route one vehicle over each of them. Second, one can disregard the time and capacity constraints of the vehicles, form a giant tour through all the nodes in the network and then partition this giant tour into a collection of routes which are feasible with regard to the capacity and time constraints of the vehicles. Generally, if there are few routes to be formed with many pick-up or delivery points in each route,then it is more effective to form a giant tour and cluster this tour into smaller segments. Note that, in making a giant tour, the nodes are ordered without regard to the cost to go from the nodes to the depot. Thus, in the case when there are many tours to be created, with a few nodes on each route, clustering nodes first will generally give better results. In this study we preferred to cluster first and then optimize each of the routes derived in the second stage.

II. THE VEHICLE ROUTING PROBLEM VARIATIONS AND SOLUTION PROCEDURES

2.1. Classification of Vehicle Routing and Scheduling Problems in General

The entries in Table 2.1. may be used to provide a quick description of the routing or scheduling problems in the literature. Taking different combinations of options within various characteristics on the left hand side of Table 2.1. results in a large number of possible problem settings.

Refering to demand as discrete or continious is done within the context of a network framework. Discrete demand implies demand located at the nodes of a network with travel between the nodes taking place along arcs. Continious demand implies that demand is spread along the arcs of the network and is satisfied for any arc by having the collection vehicle move along that arc. Uniform demand means that the demand for service at any point is the same as any other point, non-uniform demand means that the demand for service at any point may be different quantities.

Discrete and uniform problems fall into an area of scheduling theory known as Traveling Salesman Problems and research on these problems has been widespread.

If the demand is discrete but non-uniform, the single route problem may still be approached as a TSP. The multi route problem, however, becomes more complex since the number of routes to be fashioned depends on how the various nodes are clustered into individual route assigments. This problem is known as truck dispatching problem in the literature.

CHARACTERISTICS	POSSIBLE OPTIONS
1. Size of Available Fleet	one vehicle multiple vehicles
2. Type of Available Fleet	homogeneous (only one vehicle type) heterogeneous (multiple vehicle types) special vehicle types (compartmentalized, etc.)
3. Housing of Vehicles	single depot (domicile) multiple depots
4. Nature of Demands	deterministic (known) demands stochastic demand requirements partial satisfaction of demand allowed
5. Location of Demands	at nodes (not necessarily all) on arcs (" " ") mixed
6. Underlying Network	undirected directed mixed euclidean
7. Vehicle Capacity Restrictions	imposed (all the same) imposed (different vehicle capacities) not imposed (unlimited capacity)
8. Maximum Route Times	imposed (same for all routes) imposed (different for different routes). not imposed
9. Operations	pickups only drop-offs (deliveries) only mixed (pick ups and deliveries) split deliveries (allowed or disallowed)
10. Costs	variable or routing costs fixed operating or vehicle acquisition costs common carrier costs (for unserviced demands)
11. Objectives	minimize total routing costs minimize sum of fixed and variable costs minimize number of vehicles required maximize utility function based on service or convenience. maximize utility function based on customer priorities

Table 2.1. VRP in general.

If the nature of the demand is uniform and continuous, for single route problems, the Chineese Postman Problem (CPP) describes the problem of travelling all arcs of a network at least once in a continuous tour and minimizing the total distance travelled. The problem is made more difficult if some of the arcs in the network are directed rather than non-directed. Not many work on multi-route problems within this framework has been found, although it would appear to have great application for many municipal services.

Regarding to the solid waste collection problem, it is obvious that because of the size an complexity of demand it is necessary to deal with multi-route problems. Arguments may be made as to whether additional constraints are applicable.

2.2. The Traveling Salesman Problem

2.2.1. Definition of the TSP

Travelling salesman problem can be defined as follows: A salesman is to visit a group of N cities and distances from each city to every other city are known. Find the tour that the salesman should follow that will allow him to visit all the cities and return to his starting point while minimazing the total distance traveled.

Implicit in the problem formulation is the assumption that this is a single route problem. This means that the capacity of the salesman in terms of the number of cities he may visit without returning to the starting point is greater than or equal to N. If this is not true the problem becomes a multi-route problem and is called the m-salesman problem which is defined in the following manner: A common terminal where each salesman starts and finishes his tour ,and a group of N cities with known intercity distances are given. Assume that one salesman can only visit k cities (k is less than N) at one time before he must return to the terminal. Find the m tours that the salesmen must make so that all the cities are visited and the total distance is minimized.

When demand is uniform, a knowledge of the capacity of the salesman or service vehicle uniquely determines the number of demand points to be visited and the number of routes.

The TSP, while being rather simple to formulate, is actually a combinatoric problem which is quite difficult to solve. This is because of the large number of discrete feasible solutions that exist for even a moderate sized problem. Consider a 20-city TSP. For nonsymetric distance matrix there are 19 factorial possible ways that the 20 cities may be ordered on the route. This means that there are $1.2165*10^{17}$ possible feasible solutions, which can be appreciated as being a very large number by noting that if one were able on a computer to enumerate one million of these solutions per minute it would take over 27.500 years to search all feasible solutions. Thus attention turns towards heuristic methods for finding solutions to the problem.

2.2.2. Alternative Heuristic Approaches to the TSP

The algorithms discussed in this section are tour construction, tour improvement and composite procedures. Tour construction procedures generate a near optimal tour by expanding or combining subtours. Tour improvement procedures attempt to find a better tour given an initial tour. Composite procedures construct a starting tour from one of the tour construction procedures and then attempt to find a better tour using one or more of tour improvement procedures.

2.2.2.1. Tour Constraction Procedures :

In studying tour constraction procedures, the following three compenents are important :

i) the choice of an initial subtour (or starting point)
ii) the selection criterion
iii) the insertion criterion

In many cases, the initial subtour is simply a randomly chosen node or loop, but there are also alternate ways of choosing a starting subtour. The second and the third criteria highlight the fact that the selection of the next node to be inserted into the current subtour and the place where that node is to be inserted are very important. Such decisions may be made simultaneously or independent of one another (e.g., for the chepeast insertion procedure.)

1) Nearest neighbor procedure (Rosenkrantz, Stearns and Lewis) [11].

It starts with any node as the beginning of a path and finds the node closest to the last node added to path then joins this pair.

2) Clark & Wright savings procedure (Clark & Wright [12], Golden [13].

The savings algorithm is an exchange procedure in the sense that at each step one set of tours is exchanged for a better set. Initially, we suppose that every demand point is supplied individually by a separate vehicle. If instead of using two vehicles to service nodes i and j, we use only one, then we obtain a savings s_{ij} in travel distance of

$$\begin{aligned} s_{ij} &= (2\, c_{1i} + 2\, c_{1j}) - (c_{1i} + c_{1j} + c_{ij}) \\ &= c_{i1} + c_{1j} - c_{ij} \end{aligned}$$

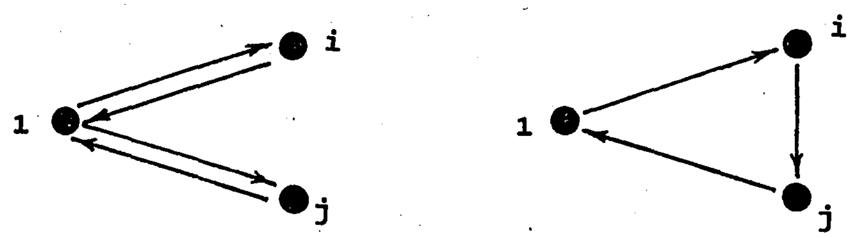


Figure 2.1. Linking nodes.

For every possible pair of tour end points i and j there is a corresponding saving s_{ij}. We order these savings from largest to the smallest starting from the top of the list we link nodes i and j with maximum savings s_{ij} unless the problem constraints are violated.

3) Insertion procedures (Rosenkrantz, Sterns and Lewis) [11]
An insertion procedure takes a subtour of k nodes at each iteration k and attempts to determine which node (not already in the subtour) should join the subtournext (the selection step) and then determines where in the subtour it should be inserted (the insertion step). For the first four insertion procedures we discuss, each node in the network can be used as a starting node in turn. Notice that when every node is used as a starting node in turn, the complexity of the entire procedure increases by an order of magnitude (that is, the number of computations is multiplied by n).

a) Nearest insertion

Given a subtour, it finds the node k, not in the subtour closest to any node, i, in the subtour, such that c_{ik} is minimal. Then, finds the arc (i,j) in the subtour which minimizes $c_{ik} + c_{kj} - c_{ij}$ and inserts k between i and j.

b) Cheapest insertion

It finds the arc (i,j) in the subtour and node k not in the subtour, such that $c_{ik} + c_{kj} - c_{ij}$ is minimal and, then inserts k between i and j.

c) Arbitrary insertion

Given a subtour , it arbitrarily selects node k not in the subtour to enter the subtour and finds the arc (i, j) in the subtour which minimizes $c_{ik} + c_{kj} - c_{ij}$. Then inserts k between i and j.

d) Farthest insertion

Same as for nearest insertion except that replace "closest to" by "farthest from" and replace "minimal" by "maximal".

e) Quick insertion (nearest addition)

This procedure starts with a single node circuit T_i, where the starting node is selected arbitrarily. Then, given the k-node circuit T_k, it finds the node z_k not on T_k that is closest to a node, calls it y_k in T_k. A new circuit T_{k+1} is constructed by inserting z_k immediately in front of y_k in T_k. This process is repeated until a Hamiltonian circuit (containing all nodes) is formed.

f) Greatest angle insertion

It finds the arc (i,j) in the subtour and node k not in the subtour such that the angle formed by the two arcs (i,k) and (k,j) is a maximum.

2.2.2.2. Tour Improvement Procedures :

The best known procedures of this type are the arc exchange procedures. (Lin [14],Lin & Kernighan [15])

In the general case, r arcs in a feasible tour are exchanged with r arcs not in that solution as long as the result remains a tour and the length of that tour is less than the length of the previous tour. Exchange procedures are referred to as r-opt procedures where r is the number of arcs exchanged at each iteration. Figure 2.2 illustrates the exchange procedures for r=2.

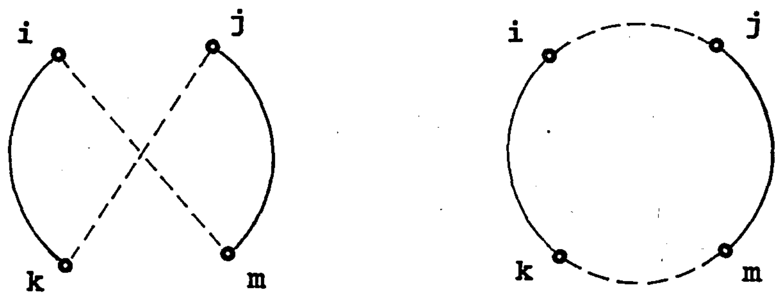
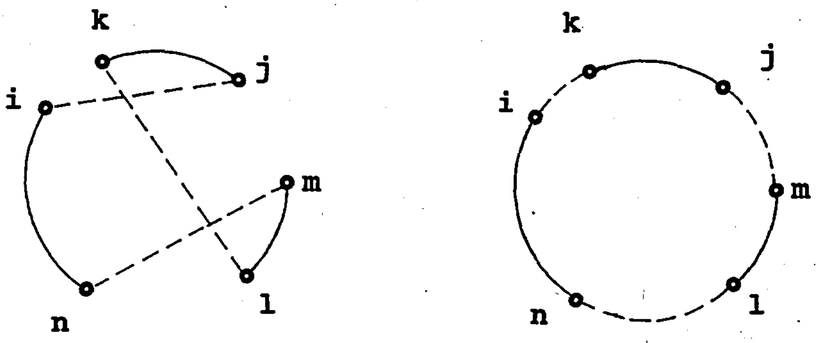


Figure 2.2. Exchange procedure for r=2

And Figure 2.3 illustrates the exchange procedure for r=3. In this example the three arcs (i,j), (l,k) and (m,n) are removed and replaced by (i,k), (j,m) and (l,n).

In an r-opt algorithm, all exchanges of r arcs are tested until there is no feasible exchange that improves the current solution. This solution is then said to be r optimal [14]. In general the larger the value of r, the more likely it is that the final solution is the true optimal. Unfortunately, the number of operations necessary to test all r exchanges increase exponentially with r.

In terms of worst case performance, Rosenkrants, Stearns and Lewis give some partial results [11]. For example, they show that a two optimal tour can be twice the length of an optimal tour assuming the triangle inequality. However, in most cases r=3 and even r=2 produce quite satisfactory results in reasonable computation times.



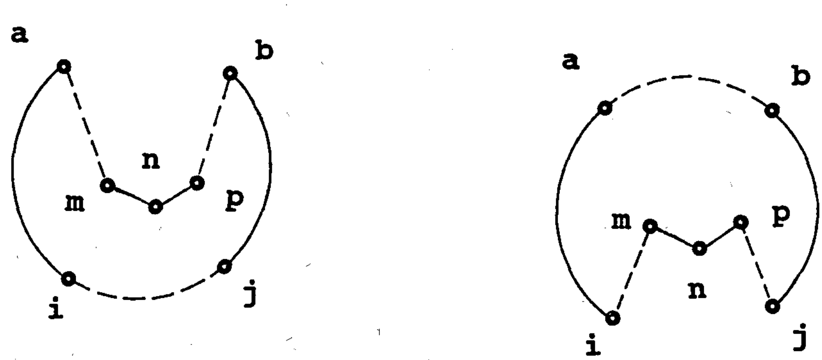
a) current tour b) tour of exchange

Figure 2.3. Exchange procedure for r=3.

As a result, values of r=2 and r=3 are the ones most commonly used. A 3-opt procedure requires n times the work that a 2-opt procedure needs. To obtain close approximations to the length of the optimal tour using this strategy, one should repeat a 3-opt procedure for a number of starting tours. Computationally, this approach may become burdensome and undesirable. Therefore, a class of heuristic methods which are computationally less demanding but as accurate as the 3-opt procedure are suggested in the literature. These are Lin & Kerninghan variable r-opt powerful algorithm [16], Or-opt procedure [17] and some composite procedures. Variable r-opt algorithm decides at each iteration dynamically what the value of r should be, namely at each iteration how many arcs to exchange. This algorithm requires considerably more effort to code than either the 2-opt or the 3-opt approach. However, it produces better solutions with fewer computations.

Recenly a modified 3-opt procedure has been proposed, that considers only a small percentage of the exchanges that a regular 3-opt would and that seems to work extremely well. This procedure, which is refered as Or-opt [17], considers only those exchanges that would result in a string of one, two or three currently consecutive nodes being inserted between two other nodes. So, by limiting the number of exchanges that need to be considered, Or-opt requires significantly fewer calculations than 3-opt. The 3-opt requires about twenty times as much computer execution time as does the Or-opt on a 100 node problem. In addition, the solutions produced by Or-opt compare very favorably with 3-opt solutions in terms of quality.

To understand how the Or-opt procedure works, we refer the reader to Figure 2.4. For each connected string of s nodes in the current tour (s = 3 first, then 2 , then 1) we test to see if the string can be relocated between two other nodes at a reduced cost. If it can, we make the appropriate changes. For s=3 in Figure 2.4., each string of 3 consecutive nodes m,n and p in the current tour is considered for insertion between all pairs of connected nodes i and j outside of the string. The insertion is performed if the total cost of the arcs to be erased (a,m), (p,b) and (i,j) exceeds the cost of new arcs to be added (i,m), (p,j) and (a,b). After considering all strings of 3 nodes, all strings of 2 nodes and then all strings of 1 node are considered. When no further exchanges improve the solution, the algorithm terminates.



a) Current tour　　　　b) Improved tour

FIGURE 2.4. An Or-exchange

2.3. Classical VRP

2.3.1. Definition

The VRP requires the determination of a set of delivery routes from one (or more) central depot(s) to various demand points, each having given service requirements, minimizing the total distance covered by the entire fleet. Vehicles have load capacities and possibly maximum route time constraints. All vehicles start and finish at central depot.We will refer to the following formulation for this problem, which was offered by Golden, Magnanti and Nguyen as general VRP [18]. If the maximum route time constraints are omitted, we obtain the standard VRP. The problem as stated is a pure delivery problem.In the case where only pick-ups are made involved, we have an equivalent problem.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{K} d_{ij} \; x_{ij}^{k} \qquad (1)$$

subject to

$$\sum_{i=1}^{n} \sum_{k=1}^{K} x_{ij}^{k} = 1 \qquad j = 2, \ldots, n \qquad (2)$$

$$\sum_{i=1}^{n} x_{ip}^{k} - \sum_{j=1}^{n} x_{pj}^{k} = 0 \qquad k = 1, \ldots, K \qquad (3)$$

$$p = 1, \ldots, n$$

$$\sum_{i=1}^{n} q_i \left(\sum_{j=1}^{n} x_{ij}^{k} \right) \le Q_k \qquad k = 1, \ldots, K \qquad (4)$$

$$\sum_{i=1}^{n} v_i \left(\sum_{j=1}^{n} x_{ij}^{k} \right) \le V_k \qquad k = 1, \ldots, K \qquad (5)$$

$$\sum_{i=1}^{n} t_i^k \sum_{j=1}^{n} x_{ij}^k + \sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}^k x_{ij}^k \leq T_k \qquad k = 1,\ldots,K \qquad (6)$$

$$\sum_{j=2}^{n} x_{1j}^k \leq 1 \qquad k = 1,\ldots,K \qquad (7)$$

$$Y_i - Y_j + n\sum_{k=1}^{K} x_{ij}^k \leq n-1 \qquad 1 \leq i \neq j \leq n \qquad (8)$$

$$Y_i,\ Y_j \in R$$

$$x_{ij}^k = 0 \text{ or } 1 \qquad \textit{for all } i, j, k \qquad (9)$$

n	:	number of nodes
K	:	number of vehicles
Q_k	:	capacity of vehicle k
V_k	:	volume of vehicle k
T_k	:	maximum time allowed for a route of vehicle k
q_i	:	demand at node i (q_1=0)
$t^k{}_1$	:	time required for vehicle k to deliver at node i
$t^k{}_{ij}$	:	travel time for vehicle k from node i to node j
d_{ij}	:	shortest distance from node i to node j
$X^k{}_{ij}$	:	$X^k{}_{ij}$ = 1 if arc i-j is traversed by vehicle k
		$X^k{}_{ij}$ = 0 otherwise

Equation (1) states that total distance covered is to be minimized. Alternatively we could minimize cost by replacing d_{ij} by a cost coefficient $c^k{}_{ij}$ which depends upon the vehicle type. Equation set (2) ensures that each demand node is served by exactly one vehicle . Route continuity is represented by equation set (3), i.e. if a vehicle enters a demand node, it must exit from that node. Equation sets (4) and (5) contain vehicle weight capacity and volume capacity constraints, similarly equation set (6) contains the total elapsed route time constraints. Equation set (7) guarantees that vehicle availability is not exceeded. Finally constraints (8) are the subtour-breaking constraints.

2.3.2. Literature References and Solution Approaches for the VRP

There are both exact algorithms and heuristic algorithms that solve the VRP as pointed out above. Since the exact algorithms can be used only for very small problems, we concentrate on heuristic algorithms, and discuss several VR heuristic methods that have been used for fairly large problems having homogeneuos fleet.

Heuristic solution strategies for VRPs can be classified as one of the following approaches :

1) Cluster first -route second procedures group or cluster nodes first and then design economical routes over each cluster as a second step. Algorithms employing this idea are the Sweep algorithm given by Gillett & Miller [19] and the algorithm by Gillett & Johnson [20] and Karp [21] for the standard single depot VRP.

The sweep approach which is an efficient algorithm for problems of upto about 250 nodes constructs a solution in two stages. First it assigns nodes to vehicles and then it determines the order in which each vehicle visits the nodes assigned to it. We select a seed node randomly. With the central depot the pivot, we start sweeping (in a clockwise or counter clockwise fashion) the ray from the central depot to the seed. Demand nodes are added to a route as they are swept. If the polar coordinate indicating angle is ordered for the demand points from the smallest to the largest, (with seeds angle = 0) we enlarge routes as we increase the angle until capacity constraints restricts us from enlarging a route by including an additional demand node.That demand node becomes the seed for the following sweep. Once we have such node clusters, we can apply TSP algorithms, such as the 2-opt, 3-opt heuristics to improve tours. In addition we can generate multiple solutions by varying the seed and select the best solution among them.

2) Route first-cluster second procedures work in the reverse sequence.First a large and usually infeasible route is constructed which covers all nodes, next this large route is partitioned into a number of smaller but feasible routes.Golden et.al.[22], Newton & Thomas [23], Bodin & Berman [24], Bodin & Kursh [25] and Stern & Dror [26] utilize this approach in their studies.

3) Savings or insertion procedures are generalizations of the similar procedures for the single route cases discussed in section (2.2.2). In these procedures we build a solution in such a way that, at each step of the procedure a current configuration that is possibly infeasible, is compared with an alternative configuration that may also be infeasible.The alternative configuration is one that gives the largest savings in terms of some criterion function, such as total cost, or the alternative configuration is one that inserts least expensively a demand entity not in the current configuration into the existing route or routes.The procedures eventually concludes with a feasible configuration.Examples of savings-insertion procedures for single depot node routing problems are described by Clark & Wright [12], Golden et.al.[18], Golden & Wong [27].

4) Improvement or exchange procedures such as the well known arc exchange heuristic developed by Lin [14] and Lin &

Kernighan [15] then extended by Christofides & Eilon [28] and Russel [29] always maintain feasibility and search for optimality. At each step, one feasible solution is altered to yield another feasible solution with a reduced overall cost and the procedure continues until no additional cost reductions are possible. These procedures are again extensions of the ones for single route problems.

5) Exact procedures for solving VRPs include specialized branch and bound, dynamic programming and cutting plane algorithms. Some of the more effective ones are described by Held & Karp [30], Hansen & Krarup [31], Balas & Christofides [32] and Crowder & Padberg [33].

6)Interactive optimization is a general-purpose approach in which a high degree of human interaction is incorporated into the problem solving process.The idea is that the experienced decision maker should have the capability of setting the revising parameters and injecting subjective assessments based on knowledge and intuition into the optimization model. This approach will be taken up in detail in the next chapter.

The first five approaches above have been used extensively in the past.The last approach represents relatively recently developed ideas.

2.4. Computer-Assisted Solution Procedure for VRP

The most impressive development from 70's to 90's in vehicle routing systems is in the computer environment. Since early systems did not have significant graphics and interactive capabilities, the user could not immediately see a suggested solution graphically and therefore could not comprehend it and react instantanously. The vehicle routing and scheduling systems developed in the beginning of 80's were the pioneers of computerized routing and scheduling. The micro computer environments, through their graphic capabilities, user friendly operating systems and software allow the user to see and understand suggested routes and interactively insert changes, make suggestions.

"In a computer assisted vehicle routing system, the user make take over the responsibility of finding a good solution from the algorithm at predefined points in the procedure and is, therefore, able tailor the solution to suit his needs.He will than use the computer power to evaluate certain characteristics of "his" solution. The solution found in this manner is called the user's solution and by definition, this solution is good." Maybe, the above statements quoted from Bodin [34] explain the importance of the user's involvement in the solution process.

The second generation micro computer based vehicle routing systems were developed in the mid 1980's. Golden, Bodin and Goodwin [35] surveyed 14 of the commercial first and second generation systems. They cost between $1000 and $150.000 and none handle all of the conditions of a real life vehicle routing problem, such as multiple vehicle types, vehicle location dependencies, single - multiple depots, time windows, and only a few of these packages survived upto now. As computers have become more potent, the user has become more demanding for high quality interactive and flexible algorithms with graphic capabilities. Some of these second generation computer assisted routing systems were designed for specific customers whose use of this systems has been well documented in the literature. Fisher [36] developed a system for Du Pont that helped reduce delivery costs 15 per cent. Underlying algorithm may be classified as a cluster first route second approach in which the clustering is directed by a mathematical programming algorithm. The implementation of a routing system for Air products and Chemicals is another application of ideas from mathematical programming by Bell [37] in which savings of 6 per cent to 10 per cent in operating cost have resulted. Brown & Grave [38] made effective use of network optimization techniques for dispatching petroleum trucks for Chevron. The system reduced transportation cost by 13 per cent even though the fleet size was allowed to increase. Evans & Norback [39] have implemented a routing system based on computer graphics for Kraft Inc. An interactive system developed on an IBM PC/XT by Belardo, Duchessi & Seagle [40] for Southland Corporation was used to construct routes for supplying the firms 7-11 stores. This system was able to save one route (out of 35) daily, thereby realizing a savings of $1000 per day. In both of these applications the underlining automatic routing algorithms were rather simple and user interaction played a substantial role. Finally, Potvin , Lapalme and Rousseau developed an interactive graphic computer system, which is called ALTO [41],[42], aimed at supporting expert algorithm designers in their task of designing and experimenting with new or already known vehicle routing algorithms on various kinds of problems.

III. CAVRS (A COMPUTER-ASSISTED VEHICLE ROUTING SYSTEM)

In this thesis a Computer Assisted Vehicle Routing System (CAVRS) similar to ALTO (Jean-Yves Potvin, Guy Lapalme and Jean-Marc Rousseau [41],[42]), in terms of its graphic multi window environment, but having additional capabilities, has been developed to support theuser in searching good solutionsto the VRP. This system is implemented on a solid waste collection and disposal problem as described in the chapter 5.

The purpose of this chapter is to describe, this interactive, graphic computer system and to present results obtained for several test problems and compare with previously published results in terms of various performance criteria. CAVR is written in Turbo Pascal and is implemented on an IBM PS/2.

3.1. Graphics and Interactive Features of CAVRS

Our work has been motivated by the fact that many problems in the field of vehicle routing are NP-hard and cannot, therefore be solved optimally using polynomaly-bounded algorithms. Hence heuristic solution strategies are widely used to solve such problems in a reasonable amount of time. For Traveling Salesman Problems (TSP), for instance, a large spectrum of heuristics have been devised. Some of these are quite robust and usually solve problems to within 2-3 per cent of optimality. However, availability, efficiency and range of application of heuristic strategies drop quickly as complexity increases. Few heuristics have been developed for constrained vehicle routing and scheduling problems (involving, for example, capacity constraints, maximum travel time constraints, etc.), and their efficiency is quite sensitive to the idiosyncratic characteristic of those problems. Expertise relating to the development of good heuristic strategies is therefore still in its infancy, and researchers in the field typically design algorithms by incrementally refining their approach. They create a first draft of an algorithm by closely analyzing the characteristics of the problems to be solved, apply the algorithm and, if the results are unsatisfactory, iteratively refine the algorithm until amenable solutions are obtained. In this context, the source code of the problem solving system has to be modified at each cycle of the iterative process, which can be a very time consuming task. Moreover, the system is likely to be inadequate for problems with some new distinctive characteristics (e.g. if capacity constraints are tighter, if vehicles are faster, etc). Motivated by such factors, CAVRS is an attempt to provide support to the user by allowing him to explore various alternative solution approaches quickly and easily when he is facing problems with specific and distinctive features.

The idea of allowing an expert user to be part of the problem solving process is not a new one and has already provided interesting results in the vehicle routing domain. However, the state of computer technology has always been a major obstacle for further progress. The breakthrough of new technologies and techniques in the field of artificial intelligence now allows OR researchers to work with hardware and software environments that enable the development of sophisticated and flexible interfaces between the user and the computerized system. Hence, it is now possible to widen the involvement of the user during the problem solving process and to allow him to dynamically orient the solution procedure as partial solutions are built up. This way, users can take advantage of their intuition and judgement while they are searching for a suitable solution. Accordingly a user friendly graphic interface is provided to the user in order to facilitate interaction with the system during the problem solving process. Figure 3.1 shows a typical CAVR screen : A transportation network with routes for vehicles

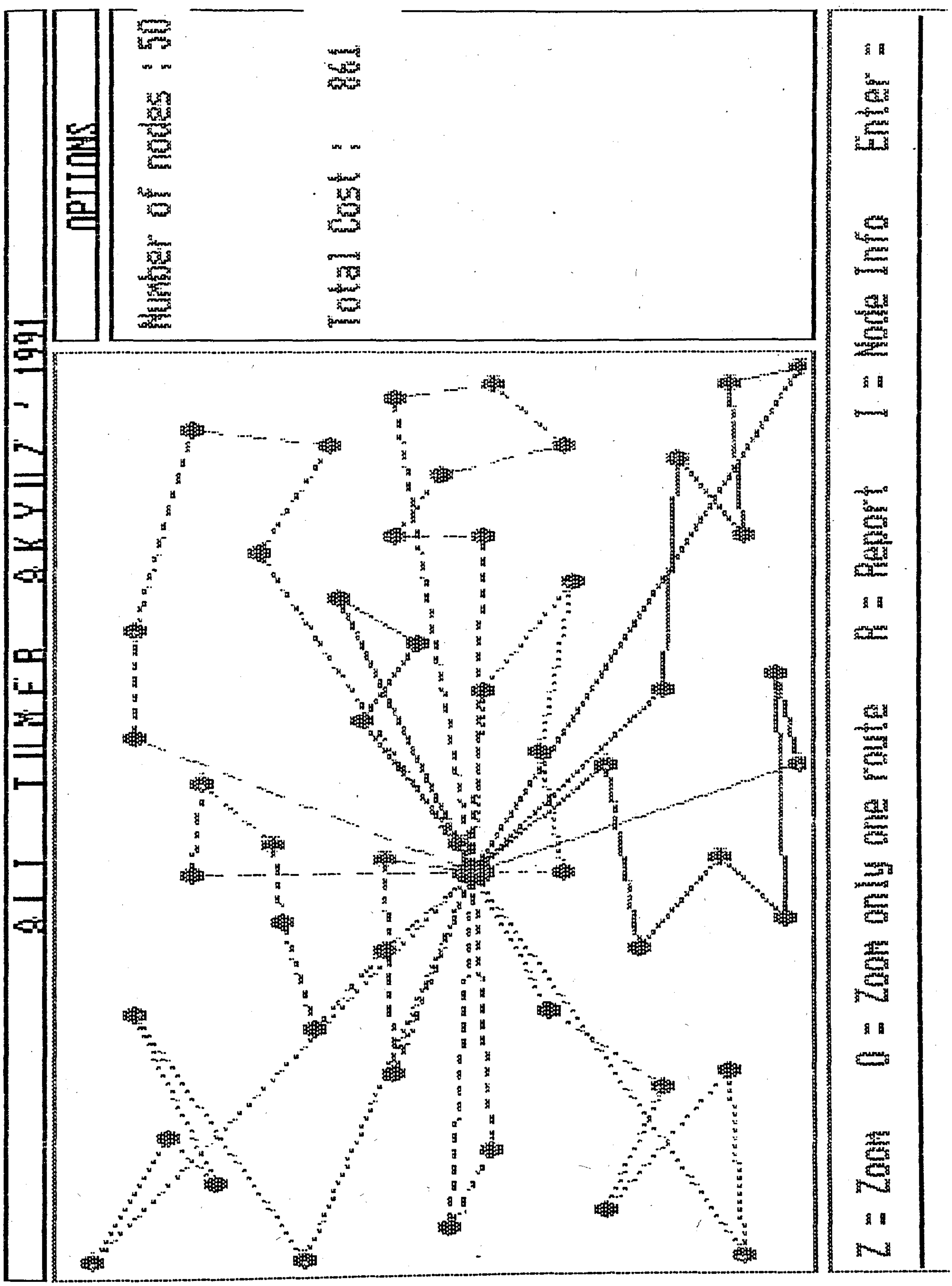


Figure 3.1. A typical CAVR screen.

is displayed in the large window on the left hand side of the screen, while the informations relating to the current routes are displayed in the window on the right hand side (total cost, route numbers etc.). The small window under the transportation network is called an "inspection window" and is used to examine various objects or entities associated with the problem.

Such a graphic, multi window environment is obviously of prime importance for computer-aided design and decision making in complex domains.In the context of vehicle routing and scheduling, this environment provides two very desirable features :

i) Interactive management of the transportation network: Transportation networks can be interactively created, edited and displayed on the CAVR screen. Partial routes can also be displayed if desired. Such graphic features are particularly important in the routing domain, because geometric and topologic consideration are important when designing routes for vehicles.

ii) Interactive problem specification: New routing problems can be easily specified via menus.For instance, it is possible to add or remove vehicles, to redefine the set of stops, to relocate the origin and to modify values of attributes (capacity of service vehicles, demands at nodes,coordinates of depot or nodes, etc.)

In conclusion CAVR offers a rich environment supporting a dynamic problem solving process where a solution is incrementally defined by the user with the help of graphic and interactive facilities.

3.2. Description of the Algorithm

3.2.1. Finding a Feasible Initial Solution to a Given VRP

In the first stage of the method, we are given a transportation network consists of (n) demand nodes and a depot. The capacity of service vehicles are also predetermined. The aim is to obtain an initial solution subject to capacity constraints, as quickly as possible.It is clear that this solution may be quite discouraging, but in this stage we only need a feasible initial solution which will be improved at later stages.

Hence, a Clark & Wright [12] saving method type procedure has been developed to quicky find an initial feasible solution to a given VRP. This method does not guarantee optimality for the vehicle routes, but it is simple to use and practical constraints can be incorporated to it quite easily. Nevertheless, the result obtained by this method become progressively worse as the constraints are made more stringent.

The algorithm has been implemented in the following fashion:

STEP 1) Initialization : (Inputting of arc, node and vehicle information)

1.1) Construct the cost matrix, C, such that:

$$C = [c_{ij}] \qquad i,j = 1,2...n$$

where n is the number of demand points plus depot.When i=j let $c_{ij} = 0$

1.2) Determine the demand of each point q_i i=1....n , the number of vehicles available T_k and the capacity of each, C_k, for each vehicle type k=1...K.

STEP 2) Construct the savings matrix and the initial solution.

2.1) Compute s_{ij} such that :

$$s_{ij} = c_{i1} + c_{1j} - c_{ij} \qquad \text{All } i,j = 2,3...n \text{ and } i \# j$$

If $s_{ij} < 0$ set $s_{ij} = 0$

If $i = j$ set $s_{ij} = 0$

2.2) Let $s_{i1} = s_{1j} = -1$ All $i,j = 2,3...n$

Note that throughout $s_{ij} = -1$ indicates the presence of (i,j) in a current solution.

STEP 3) Determine a candidate pair.

3.1) Find the ordered pair i^*,j^* with the greatest feasible savings such that :

$$s_{i*,j*} = \text{Max} [s_{i,j}]$$

where (i,j) is defined over all ordered pairs such that :

$$s_{i1} \# 0 \quad \text{and} \quad s_{1j} \# 0$$

3.2) If $s_{i*j*} = 0$ go to STEP 5.

STEP 4) Join the points $i*$ and $j*$ on a route.

4.1) If neither of the points is on a route, construct a new route Z and compute the required demand Q_Z such that:

$$Q_Z = q_{i*} + q_{j*}$$

Go to STEP 4.4.

4.2) If one of the points is currently assigned to a route, say Z, attempt to join the unassigned point to Z. Compute the total demand Q_z such that:

$$Q_z = Q_z + q_i$$

4.3) If both points are currently assigned to routes, say U and V. Attempt to join both routes into one route Z. Compute the total demand Q_z such that:

$$Q_z = Q_u + Q_v$$

4.4) Check the capacity restrictions. Select the smallest C_k such that;

$$C_k \geqslant Q_z$$

and proceed to 4.5. If no such C_k exists set $s_{i*j*} = 0$ and return to step 3.

4.5) Update the number of vehicles available, T_k, such that:

$$T_k = T_k - 1$$

If routes are joined, increment the appropriate number of vehicles available for the previous truck size used on this route.

4.6) Update the savings matrix such that:

$$s_{i*j*} = -1 \qquad s_{j*i*} = 0$$
$$s_{i*1} = 0 \qquad s_{1j*} = 0$$

4.7) If there is still a candidate pair, go to Step 3.

STEP 5) Save the solution.

5.1) Maintain the routes formed and the order in which points were joined.

Sample Problem :

Consider a small problem involving six demand points and a single terminal. The cost matrix is given by C in Table 3.1. as is the list of demands for all points. It is assumed that there is one 16-unit capacity vehicle available and an unlimited number of 8-unit capacity vehicles.

Data for Sample Problem :

Cost matrix and demand requirements are,

	1	2	3	4	5	6	7
1	0	20	30	50	60	50	40
2	10	0	5	10	20	20	15
3	20	10	0	30	10	35	20
4	30	15	20	0	10	15	10
5	40	15	5	10	0	15	5
6	30	30	25	10	5	0	20
7	20	10	30	20	10	30	0

Point #	Demand
1	0
2	6
3	2
4	5
5	5
6	8
7	6

Table 3.1. Cost matrix and demands (point#1 is depot)

Applying the Algorithm :

STEP 1 and 2) From the cost DATA, the savings matrix can be constructed as in Table 3.2. For example, the savings between the Node 2 and Node 4 is calculated such that :

$$s_{24} = c_{21} + c_{14} - c_{24}$$

$= 10 + 50 - 10$
$= 50$

All s_{i1} and s_{1j} are set at (-1), indicating the initial set of routes where every demand points is serviced, although inefficiently, by a single vehicle moving from the terminal to the point and directly back.

STEP 3) The maximum element in S is found to be s_{65} with a savings of 85. Hence points 6 and 5 are candidates for inclusion in a single route.

	1	2	3	4	5	6	7
1	0	-1	-1	-1	-1	-1	-1
2	-1	0	35	50	50	40	35
3	-1	30	0	40	70	35	40
4	-1	35	40	0	80	65	60
5	-1	45	65	80	0	75	75
6	-1	20	35	70	85	0	50
7	-1	30	20	50	70	40	0

Table 3.2. Initial savings matrix for sample problem.

STEP 4) Since neither point 6 nor point 5 are assigned to a route, the total demand is computed for a possible new route as $q_5 + q_6 = 13$. Since there is a vehicle of sufficient capacity available to accommodate the demand, the two points are joined.

In the current savings matrix,

$s_{65} = -1$ $s_{56} = 0$ $s_{61} = 0$ $s_{15} = 0$.

STEP 3) The procedure returns to the current updated savings matrix and element s_{54} is seen the maximum "savings" element with :

$$\text{Max } s_{ij} = s_{54} = 80$$

STEP 4) Point 5 is currently on a route such that the demand of the route when augmented necessarily by the demand of point 4 becomes :

$$13 + 5 = 18$$

which exceeds the capacity of the largest available vehicle.Hence s_{54} is set equal to zero in the current savings matrix and a new candidate pair is considered.

The procedure continues in similar fashion investigating maximum remaining savings. The first pair found to be feasible is the pair (5,3) with s_{53} = 65. All other point pairs are found to be infeasible and the final routes are :

(1-2-1)
(1-4-1)
(1-7-1)
(1-6-5-3-1)

serviced by three 8-unit vehicles and a single 16-unit vehicle respectively. Since $s_{i*j*} = 0$, the procedure moves to step 5 and is terminated.

3.2.2. Single Route Improvements Stage

In this stage, there are m routes and m different Traveling Salesman Problems to be solved seperately. Each individual routing problems considered has capacity constraints on the service vehicles. We will show how local search can be efficiently implemented in these situations. Our main motivation is for the development of k-exchange procedures that can handle side constraints efficiently. This is because in many real life routing problems, including solid waste collection there are various side constraints. (Croes,[43]; Lin,[14]; Lin & Kerninghan,[15].)

As mentioned in section 2, a two-exchange involves the substitution of two edges, say (i, i+1) and (j, j+1) in the current solution, with two other edges (i,j) and (i+1, j+1) not in the current solution (see Figure 3.2.) Note that the orientation of the path {i+1,.....j} is reversed in the new route.

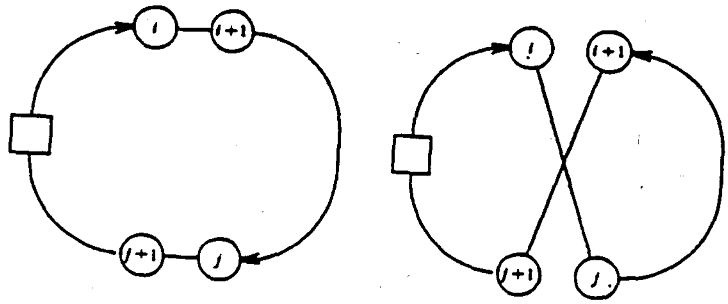


Figure 3.2. A two-exchange.

Such an exchange results in a local improvement if and only if :

$$c_{ij} + c_{i+1,j+1} < c_{i,i+1} + c_{j,j+1}$$

Therefore testing an improvement involves only local information and can be done in constant time.

In a two-exchange the two edges (i, i+1) and (j, j+1) that will be deleted, uniquely identify the two edges (i, j) and (i+1, j+1) that will replace them. However in a three-exchange, where three edges are deleted, there are several possible ways to construct a new route from the remaining segments. Figure 3.3. shows two possible three-exchanges that can be performed by deleting the edges (i,i+1), (j, j+1) and (k, k+1) of a route.

For all possibilities, conditions for improvements are easily derived.There is one important difference between the two there-exchanges shown in Figure 3.3. : In one case the orientation of the path {i+1,.....j} and {j+1,.....k} is reversed whereas in the other case this orientation remains the same as the original route.

Since the computational requirement of 3-optimality becomes prohibitive if the number of nodes increases, it is prefered to use 2-exchange procedure rather than 3-exchange procedure at the stage of single route improvement.

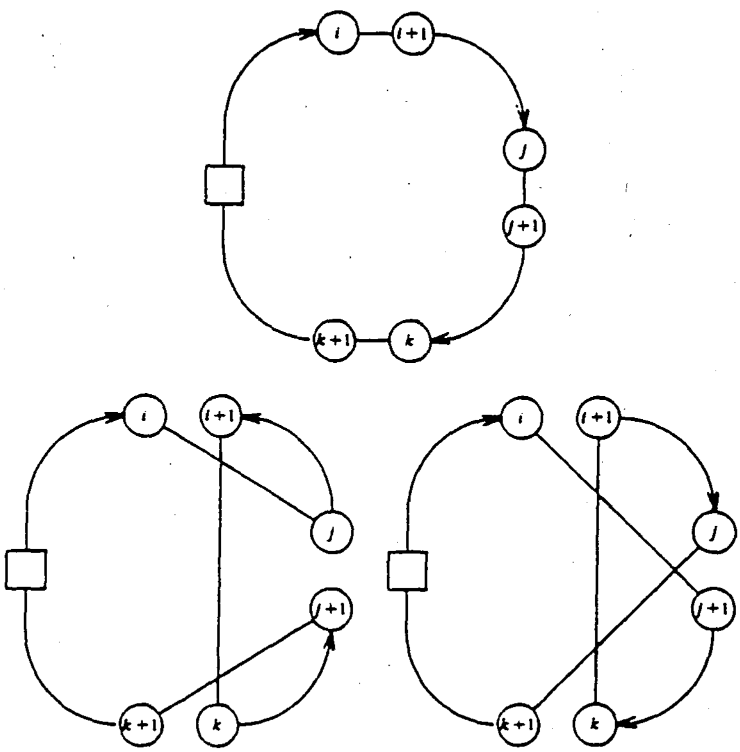


Figure 3.3. Two ways to perform a 3-exchange.

One important problem with the use of k-exchange procedures in the TSP with side constraints is testing the feasibility of an exchange. A 2-exchange, for instance, will reverse the path {i+1,......j}, which means that one has to check the feasibility of all the nodes on the new path with respect to those constraints. In a straightforward implementation this requires T(n) time for each 2-exchange, which results in a time complexity of T(n3) for the verification of 2-optimality.

It is assumed that the current route is given by a sequence (1,......i,......n) where i represents the ith node of the route, 1 represents the origin and n represents the last node of the routes. It is also assumed that we are always examining the exchange that involves the substitution of edges (i, i+1) with (j, j+1) and (i, j) and (i+1, j+1) in a 2-exchange operation.

We choose the edges (i, i+1) in the order in which they appear in the current route starting with (1, 2). After fixing an edge (i, i+1), we choose the edge (j, j+1) from the set { (i+2, i+3), (i+3, i+4)......(n-1, n) } in the given order, and calculate the savings for all possible exchanges for this fixed edge (i,i+1).

Now that search and exchange order have been determined, let us return to the feasibility question. If some arcs in the network are directed or some connections between nodes are not allowed, changing the order of nodes in a route after a 2-exchange operation may disrupt feasibility of current solution. In order to maintain feasibility a control mechanism is required at each 2-exchange step.

A step by step detailed description of the above explained procedure is as follows :

STEP 1 : A route given as (1,2,....,n)

STEP 2 : 2.1. i=1 to n-1
2.2. j=i+2 to n

STEP 3 : 3.1. if (i < n) and (j < n) then calculate the single save:

$$s = (c_{ij} + c_{i+1,j+1}) - (c_{i,i+1} + c_{j,j+1})$$

3.2. if (i < n) and j = nthen calculate the single save :

$$s = (c_{ij} + c_{i+1,1}) - (c_{i,i+1} + c_{j,1})$$

STEP 4 : 4.1. If single save < 0, check feasibility then perform exchange and update the route.

4.2. Calculate new total cost.

4.3. Go to STEP 2.

STEP 5 : A route that is 2-optimal.

STEP 6 : Replay the algorithm for all routes.
Goto STEP 1.

3.2.3. Node Interchange Procedures Between Routes

In the first stage, a feasible solution is found to a VRP by using a saving approach and m different routes serviced by m vehicles, are obtained. In the second stage, these routes are considered separately as a bunch of single vehicle TSP and an effort is made to decrease the total travel cost on each route by using a heuristic improvement technique based on the idea of arc interchanges.
At the end of the second stage there exist m routes which are feasible and are individually 2-optimal.However, there is still a chance to improve this solution to the VRP by considering inter route node (and arc) exchanges. The goal of this improvement stage is to reduce the total distribution cost as reflected by our objective function, while maintaining route capacity constraints. The procedures that attempt to accomplish that are based on what is called the Node Interchange Procedure [44].Let us now describe in detail the Node Interchange Procedure, together with some conceptual differences between this procedure and arc interchange heuristics that need to be clarified. (The most commonly known examples of the latter are the two-opt, three-opt, r-opt and Or-opt procedures. A detailed description of these can be found in the previous sections of this study.)

The Node Interchange Procedure is a two-node rather than two-arc interchange heuristic. In addition, unlike the two-opt

$$s_{[i,j\,:\,k,r,l]} = c_{ij} + c_{kr} + c_{rl} - c_{jr} - c_{ri} - c_{kl}$$

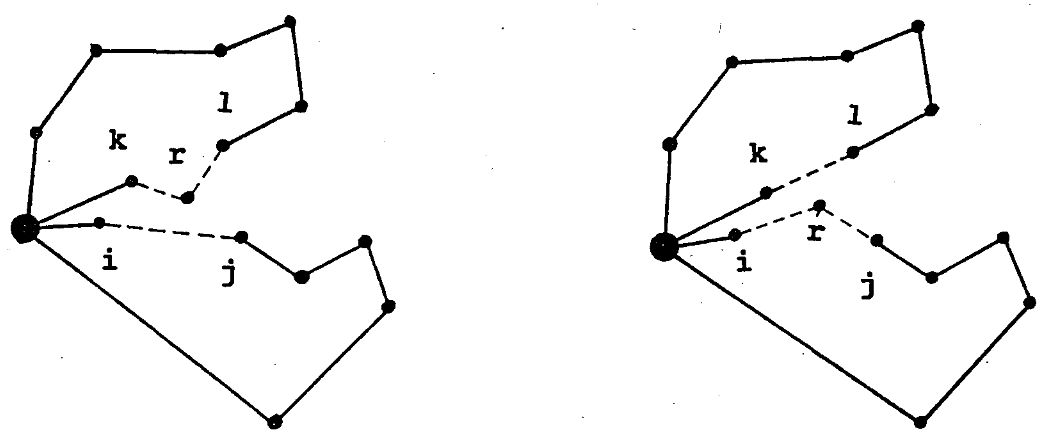


Figure 3.4. A one-node interchange operation.

$$s_{[i,r,j:k,s,l]} = c_{jr} + c_{ri} + c_{ks} + c_{sl} - c_{kr} - c_{rl} - c_{js} - c_{si}$$

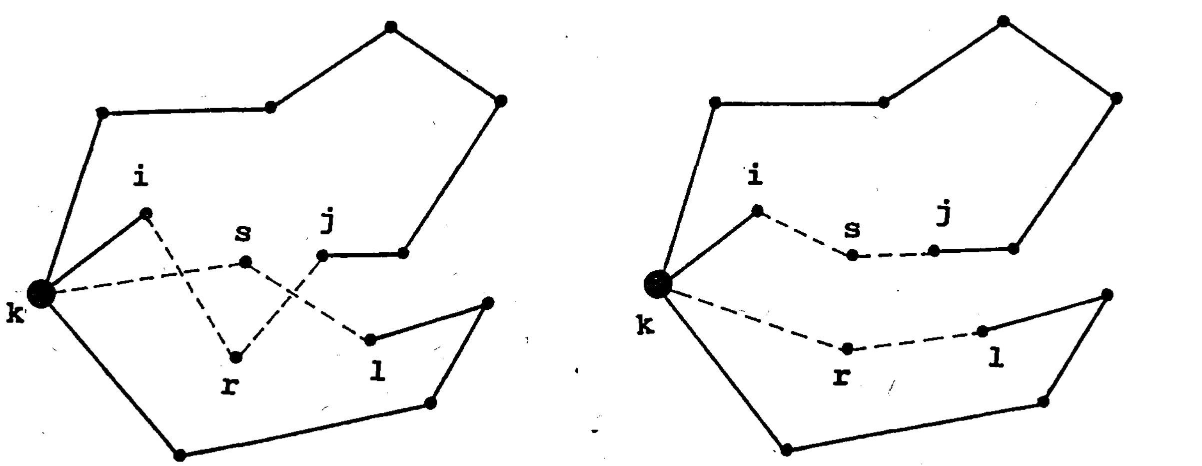


Figure 3.5. A two-node interchange operation.

heuristic, it is not limited to a single salesmen tour. In order to apply the two-opt heuristic to a given m-traveling salesmen tour a transformation of the m tour to an equivalent 1 tour is necessary. Given m>1 tours, we say that the m tours are two-swap optimal if it is impossible to obtain an improved m tour solution by exchanging the positions of one or two nodes. We illustrate a single one node interchange operation in Figure 3.4., and a single two-node interchange operation in Figure 3.5.

If the two-node interchange is applied to a single route, then it is equivalent to a four-arc exchange. If one-node interchange is applied to a single route, then it is equivalent to a three-arc exchange. Lin found that for the TSP the four-exchange heuristic did not noticeably better solutions than the three-opt solution [14]. The strength of the node-interchange procedure is its applicability to an m-tour solution for $m > 1$. For a given solution to a VRP, the candidate nodes for an interchange must have a strictly positive savings term and must satisfy the individual tour constraints as well.

The Calculation of the Saving Terms and Feasibility Checks :

Assume that the number of vehicles available is W. Denote by P_{rh} the subset of costumers that constitute the route r for the vehicle h. Given a feasible set of routes for the planning period, in order to transform this set into a set of routes that are two-swap optimal, two different node interchange operations are employed :

i) One node interchange between routes.
ii) Two node interchange between routes.

Each of the node interchanges has a different expression that computes the savings in the cost.

i) One-node interchange savings :

$$s\,[i,j : k,r,l] = c_{ij} + c_{kr} + c_{rl} - c_{jr} - c_{ri} - c_{kl}$$

This term expresses the savings if node r switches routes. Its original routing position is between nodes k and l, and a subsequent routing position would be between nodes i and j. (See Figure 3.4)

ii) Two-node interchange savings :

$$s\,[i,r,j : k,s,l] = c_{jr} + c_{ri} + c_{ks} + c_{sl} - c_{kr} - c_{rl} - c_{js} - c_{si}$$

This term expresses the saving if node r and s changes their routes. Node r's original routing position is between nodes j and i, and node s's original routing position is between nodes k and l. A subsequent routing position would be node r between nodes k and l, and node s between nodes j and i. (See Figure 3.5.)

It is clear that, not all favorable trade-offs -as indicated by a positive sign on the saving term value- are feasible if we check the corresponding vehicles' capacities, for the proposed interchange. So, we have to make sure that, the vehicle capacity will not be exceeded when interchanging nodes among the respective routes. In the one-node interchange the following conditions have to be satisfied before node r changes its route:

i) The savings term S has to be positive.
ii) If we denote the amount of demand at node r by d_r, and the capacity of the n^{th} vehicle by q_n,
(Total previous demand on route 1) + $d_r \leqslant q_n$

In the two-node interchange the following conditions have to be satisfied before nodes r and s are swapped in their respective routes :

i) The savings term S has to be positive.
ii) (Total previous demand on route 1) - $d_r + d_s \leqslant q_1$
iii) (Total previous demand on route 2) - $d_s + d_r \leqslant q_2$

The Node Interchange Procedure :

We now describe one-node and two-node interchange procedures in detail.The initial input to both procedures is the set of vehicle routes which we have obtained as output from the two-opt arc exchange procedure. Each of the interchange procedures at some point implement a single route improvement heuristic.

In this algorithm we find the feasible node interchange which results in the greatest improvement in the value of the objective function and implement this node interchange. We then recompute all node interchanges and list the ones which are feasible and improve the value of the objective. Interchanges in this procedure are implemented one at a time. When there are no more feasible exchanges having a positive savings term , the procedure attempts to reoptimize each route using an arc exchange algorithm, namely, two-opt procedure. After reoptimizing each route, further feasible node interchanges with positive savings are searched and the criteria for termination is satisfied when no such interchanges are found.

A step by step detailed description of the above explained procedure is as follows :

STEP 1) For i=1.....m implement Two-opt [route(i)]

STEP 2) Compute the savings terms and check feasibility for all one-node and two-node interchange transformations. Drop non-positive savings and/or infeasible interchanges.If there are no positive and feasible savings stop.

STEP 3) Sort the savings term in non-increasing order.

STEP 4) If the list of the savings terms is not empty, perform the interchange transformation that corresponds to the first savings term. If the list empty Go to STEP 1.

STEP 5) Recompute the other savings terms on the list while dropping the non-positive and/or infeasible terms. Go to STEP 3.

This node interchange operation for a given set of vehicle routes is similar to the arc exchange transformation on a single route [44]. This concept of node interchange operation on routes is instrumental in designing improvement schemes with the flexibility of interchanging the positions of two nodes, one node, or even deleting or inserting nodes into a vehicle routing solution. When interchanging the positions of nodes on a route and between routes, node characteristics ,such as demand and physical locations, are kept the same. Given an approximately uniform distribution of nodes over a bounded geographic area with the depot in its center, and a randomly partition of these nodes to individual routes, a node interchange operation between initially constructed routes proved to be a very effective tool for redesigning an improved routing system. But we preferred, before applying node interchange operation between routes, to perform a single route improvement heuristic, namely, two-opt procedure in order to improve the initial set of routes.

3.2.4. Or-Opt Procedure for the Final Improvements

At this stage we have again m-single traveling salesman routes, which are all two-optimal and two-swap optimal. Now, in order to improve this solution we need a more powerful heuristic, such as three-opt procedure. Because the computational requirement to verify three optimality becomes prohibitive if the number of nodes increases, we prefered to use the Or-Opt procedure, which, as explained in section (2.2.2), is nearly as powerful.

Actually, the procedure that we used is slightly different from the original Or-Opt algorithm. Firstly, in Or-Opt procedure, for each connected string of S nodes in the current tour (S equals three first, then two, then one), we test to see if the string can be relocated between two other nodes at a reduced cost. Here we relocated a string of two consecutive nodes first (S=2), then only one node (S=1) between two others. In other words, we never consider a string of three consecutive nodes to relocate on a route. Furthermore, when an insertion is performed, algorithm goes back to the first step whereas, in original algorithm, when an insertion is performed, it continues to search forward for possible new Or-exchanges.

We denote OR-2 as an Or-exchanges for S=2, and OR-1 as an Or-exchanges for S=1 as can be seen that in Figure 3.6. and Figure 3.7.

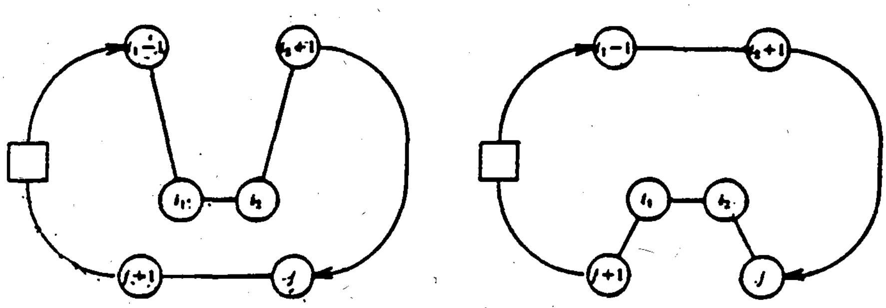


Figure 3.6. An OR-2 exchanges

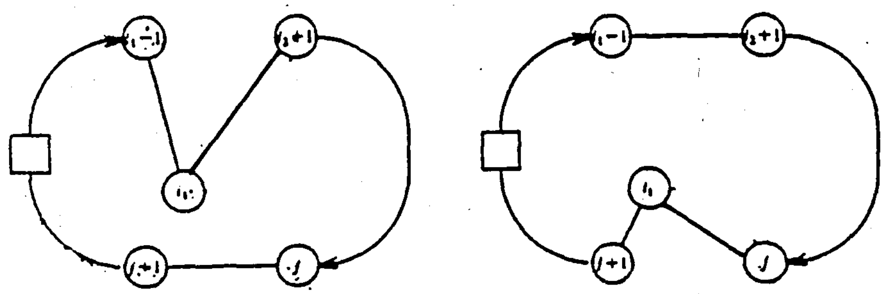


Figure 3.7. An OR-1 exchanges

A step by step detailed description of this procedure is as follows :

STEP 1-a : Consider every edge (i_1,i_2) lying between i_1-1,i_2+1 in the existing tour. Temporarily add (i_1-1,i_2+1) to and delete (i_1-1,i_1), (i_2,i_2+1) from the tour, and replace one by one every edge $(j,j+1)$ in the path from i_2+1 to i_1-1 by the chain $\{(j,i_2),(i_2,i_1),i_1,j+1)\}$.

STEP 1-b : If an improvement is found the change is made permanent. Go to step 1-a.

STEP 2-a : Consider every node i_1 lying between nodes i_1-1,i_2 in the existing tour. Temporarily add (i_1-1,i_2) to and delete (i_1-1,i_1), (i_1,i_2) from the tour, and replace one by one every edge $(j,j+1)$ in the path from i_2 to i_1-1 by the chain $(j,i_1),(i_1,j+1)$.

STEP 2-b : If an improvement is found, the change is made permanent. Go to step 2-a.

STEP 3 : When no further OR-2 and OR-1 exchanges improve the solution the algorithm terminates.

STEP 4 : Take another route. Go to STEP 1.

It can easily be seen that, we don't care about the capacity constraints. Because, in this stage we have a bunch of routes and we attempt to obtain an improvement on these routes independently. So, the total demand on any route does not increase or decrease and since the previous solutions are all feasible, final solutions are also feasible.

IV. COMPUTATIONAL PERFORMANCE OF CAVRS

4.1. Literature Problems for the VRP

In practice it is difficult to find the best method for an actual problem. In the literature some data with coordinates are published which allows to measure the quality of several methods. A summary of the data sets most frequently used in the literature for algorithm performance evaluations are given in Table 4.1. and Table 4.2. , Table 4.3. shows the results of the different methods. Only cases are included in this analysis for which numerical results are available in the literature.

Name	Problem Source	# of nodes	Depot Coordinates	Vehicle Capacity
P1	Christofides & Eilon [28]	50	(30,40)	160
P2	Christofides & Eilon [28]	75	(40,40)	140
P3	Christofides & Eilon [28]	100	(35,35)	200
P4	Gillette & Miller [19]	75	(40,40)	**100**
P5	Gillette & Miller [19]	75	(40,40)	**180**
P6	Gillette & Miller [19]	75	(40,40)	**220**
P7	Gillette & Miller [19]	100	**(40,50)**	**112**
P8	H.Paessens [45]	50	(30,40)	80
P9	H.Paessens [45]	50	(0, 0)	160
P10	H.Paessens [45]	100	(35,35)	100

Table 4.1. Literature Problems.

Three of the problems proposed by Christofides and Eilon [28] along with seven variations are presented in Table 4.1. Problems P4, P5 and P6 are the same as problem P2, except that the capacity constraints are changed. The same relation holds between Problem pairs P3 , P10 and P1, P8. Problem P9 is the same as problem P1, except that the depot coordinates are changed. Finally, problem P7 is the same as problem P3 , except that depot coordinates and capacity constraints are changed. This was done to illustrate that the time to solve a given problem is highly dependent on the average number of locations per route and much less on the total number of locations.

4.2. Computational Experience

A computer program was developed to implement the CAVR algorithm. The code was written in Turbo Pascal and executed on an IBM PS/2.

In order to test the computational performance of the CAVR algorithm in relation to other heuristics, it was implemented on the above widely tested set of vehicle routing problems .

The computational information is reported in Table 4.3. While the total distances travelled are of concern, the number of vehicles utilized in all cases is also important to note. Moreover, it should be pointed out that no attempt has been made to convert computing times to comparable values. Hence, caution should be exercised in viewing solution times.

METHOD 1 :	Christofides & Eilon	/ Savings Method
METHOD 2 :	Christofides & Eilon	/ 3-Optimal Tour
METHOD 3 :	Gillette & Miller	/ Forward Sweep
METHOD 4 :	Holmes & Parker	/ Suppression
METHOD 5 :	Gillette & Johnson	/ Modified Sweep
METHOD 6 :	Beltrami & Bodin	/ Cluster Second
METHOD 7 :	Beltrami & Bodin	/ Cluster First
METHOD 8 :	Gaskell	/ Sequential Saving
METHOD 9 :	Paessens	/ Parellel Saving 1
METHOD 10 :		/ Parellel Saving 2
METHOD 11 :	Probol	
METHOD 12 :	Heins	/ Two Phase
METHOD 13 :	Christofides	/ Tree Search
METHOD 14 :	Neitzel	
METHOD 15 :	Mole & Jameson	
METHOD 16 :	Fischer & Jaikumar	

Table 4.2. Descriptions of methods on Table 4.3

	P1			P2			P3			P4			P5			P6			P7			P8			P9			P10		
*	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME	SOL.	VEHI.	TIME
METHOD 1	585	6	0.6	900	10	1.3	887	8	2.5																					
METHOD 2	556	5	2	876	10	4	863	8	10																					
METHOD 3	574	5	2	865	10	1.23	864	8	6	1127	15	0.68	765	8	2.23	723	7	3.68	1176	14	1.83									
METHOD 4	573	5	0.58	886	10	2.42	876	8	2.04																					
METHOD 5	524	5	0.06	865	10	0.09	851	8	0.3	1096	15	0.08	752	8	0.15	704	7	0.17	1146	14	0.13									
METHOD 6	579	6	0.05	901	11	0.11	864	8	0.2	1054	15	0.11	786	8	0.11	737	7	0.11	1136	14	0.18									
METHOD 7	636	6	0.08	958	11	0.18	969	8	0.48	1142	15	0.18	870	8	0.18	786	7	0.18	1263	15	0.48									
METHOD 8	625	5		1001	11		1008	8																						
METHOD 9	585	6		907	10		889	8														791	11		832	5		1241	15	
METHOD 10	580	6		905	11		869	8																						
METHOD 11	525	5		859	10		843	8														812	11		844	5		1266	16	
METHOD 12	537	5		873	11		858	8														774	11		836	5		1241	15	
METHOD 13	547	5		883	11		851	8																						
METHOD 14	533	5		888	10		897	8																						
METHOD 15	575	5		910	10		882	8																						
METHOD 16	524	5		857	10		833	8																						
AVERAGE	566	5.31	0.77	896	10.4	1.33	854	8	3.07	1105	15	0.26	793	8	0.67	736	7	1.04	1180	14	0.66	792	11		837	5		1249	15.3	
CAVR	567	6	0.41	868	11	1.08	857	8	2.5	1053	16	0.75	763	8	0.83	717	7	0.92	1238	14	1.17	769	11	0.28	822	5	0.38	1205	16	1.17
▪ BEST	8.20			1.20			2.80			0.00			1.40			1.80			8.90			0.00			0.00			0.00		
▪ AVE(%)	0.18 (-)			3.23 (+)			0.35 (-)			4.94 (+)			3.93 (+)			2.65 (+)			4.92 (-)			2.99 (+)			1.82 (+)			3.65 (+)		

* Method descriptions are given in Table 4.2

Table 4.3. Results of Literature Problems by Various Methods

The CAVR algorithm is able to produce better solutions for the larger N/U values where, N is the minimum number of routes and U is the average number of nodes per route. The larger the value N/U the better the results for the CAVR method, especially in case of centeralized depot.(see Table 4.4.). The CAVR solutions for problems P4, P8, P9 and P10 are the best available solutions; and for problems P2, P3, P5 and P6, the solutions by this method are very close to the best available solutions:

For problems P2 and P3; the best solutions which were attained by the method developed by Fisher and Jaikumar, are only 1.2 per cent and 2.8 per cent better, while the CAVR solution of P2 is 3.2 per cent better and that of P3 is 0.35 per cent worse than the average of the 16 different methods considered in the comparison.

For problems P5 and P6; Modified Sweep Algorithm of Gillette and Johnson gave the best results with 1.4 per cent and 1.8 per cent differences respectively. The CAVR algorithm generates 3.9 per cent and 2.6 per cent better results than the average of four methods used in the comparison.

For problems P1 and P7, the Sweep Algorithm is able to produce better solutions than CAVR, because the smaller the value N/U the better the results for the sweep method. The variances between the best solutions and the CAVR results are 8.2 per cent and 8.9 per cent respectively. However; the average of 16 methods used for comparison is only 0.17 per cent better for P1, while the difference between the average of four methods and CAVR algorithm is 4.9 per cent on behalf of the average. Table 4.5. tabulates the percent improvements in all stages during the solution process of the problems cited above by CAVR.

PROBLEM NO.	NUMBER OF NODES	MAX. LOAD	DEPOT COORD.	CENTRAL/ DECENT.	NODES PER ROUTE	MIN.NO. OF ROUTES	N / U**
P1	50	160	(30,40)	CENTRAL	10.00	5	0.50
P2	75	140	(40,40)	CENTRAL	7.50	10	1.33
P3	100	200	(35,35)	CENTRAL	12.50	8	0.64
P4	75	100	(40,40)	CENTRAL	5.00	15	3.00
P5	75	180	(40,40)	CENTRAL	9.37	8	0.85
P6	75	220	(40,40)	CENTRAL	10.71	7	0.65
P7	100	112	(40,50)	CENTRAL	7.14	14	1.96
P8	50	80	(30,40)	CENTRAL	4.54	11	2.42
P9	50	160	(0 , 0)	DECENTRAL	10.00	5	0.50
P10	100	100	(35,35)	CENTRAL	6.66	15	2.25

** N is the minimum number of the routes.

U is the average number of routes per route.

Table 4.4. Characteristics of the problems.

PROBLEM	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
INITIAL	859	1105	1301	1201	1074	1087	1550	861	1224	1617
% CHANGE	32.48%	18.37%	30.05%	12.07%	26.44%	30.54%	19.74%	10.10%	30.64%	25.42%
STAGE 2	580	902	910	1056	790	755	1244	774	849	1206
% CHANGE	2.24%	3.77%	4.29%	0.28%	3.29%	3.58%	0.00%	0.00%	3.18%	0.17%
STAGE 3	567	868	871	1053	764	728	1244	774	822	1204
% CHANGE	0.00%	0.00%	1.61%	0.00%	0.13%	1.92%	0.48%	0.65%	0.00%	0.00%
FINAL	567	868	857	1053	763	714	1238	769	822	1204

Table 4.5. Improvement In Stages

V. APPLICATION OF CAVR TO HAZARDOUS SOLID WASTE COLLECTION IN ISTANBUL

5.1. Introduction

In this section , a study on vehicle routing modelling of the collection and transportation system for the hazardous solid waste of hospitals in Istanbul metropilitan area is presented. In order to determine the capacities and detailed routes of the vehicles used in the collection of hazardous solid wastes of these hospitals, CAVR algorithm have been implemented. Within this context numerous transportation alternatives have been examined so as to improve the total solid waste transportation cost. As number and location of the disposal sites had to be predefined for this study, the results of the Master thesis by Akgul [46] were used for that purpose and Karakiraz, which is due North-East of Marmara region and near the Black sea Coast, was chosen.

It was decided that the overall problem giving due consideration to all disposal sites and transfer station locations, number of vehicles and vehicle routes simultaneously, would create an unnecessarily huge and complicated model. Therefore, the problem was divided into two stages. The first stage in which the detailed routes of vehicles are ignored and decisions related to the location of disposal sites are investigated, is already accomplished by Akgul [46]. The second stage in which the detailed routes and the fleet sizes are determined according to predetermined hospital and disposal site locations is taken up in this study. In both stages, minimization of total solid waste collection and transportation cost was the objective.

As all the decisions variables are not be simultaneously handled within the summarized two-stage approach, the obtained results can not be guaranteed to be the best solutions that make the total cost minimum. However because of the high disposal-site costs which are independent of routing considerations and high transportation-time to collection-time ratios, it is assumed that predetermining the site location will not affect the results very much.

In the micro approach which is the second stage, vehicle routings for the collection system is attained by using the CAVR algorithm introduced and developed in section 4.

5.2. Data Compilation

Data used in this study have been compiled from different sources: Traffic Administration of Istanbul, reports of Bogazici University and Istanbul Technical University Transportation Planning Projects, Village Services Istanbul City Management Publications, Ministry of Public Health Publications and other publications which have been prepared for Istanbul.

One major problem has been the lack of sufficient information about the hazardous solid wastes produced at the hospitals in Istanbul metropolitan area. No statistics were available about type and volume of such hazardous material. Knowledge about the routes used for hazardous material transport is also inexact. In determining the amount of the hazardous solid wastes produced, the existing number of beds and number of surgical operations at hospitals are used as a basis, according to the Ministry of Public Health Publications [47]. According to existing statistics obtained from Hospitals in Izmir , the amount of hazardous solid waste per bed is normally 0,160 kg/day. We assumed an additional amount of 2 kg hazardous waste per surgical operation and another 20 per cent for unexpected events.

In this case, expected demand at node i, can be calculated by using the equation, in terms of (kg/week) :

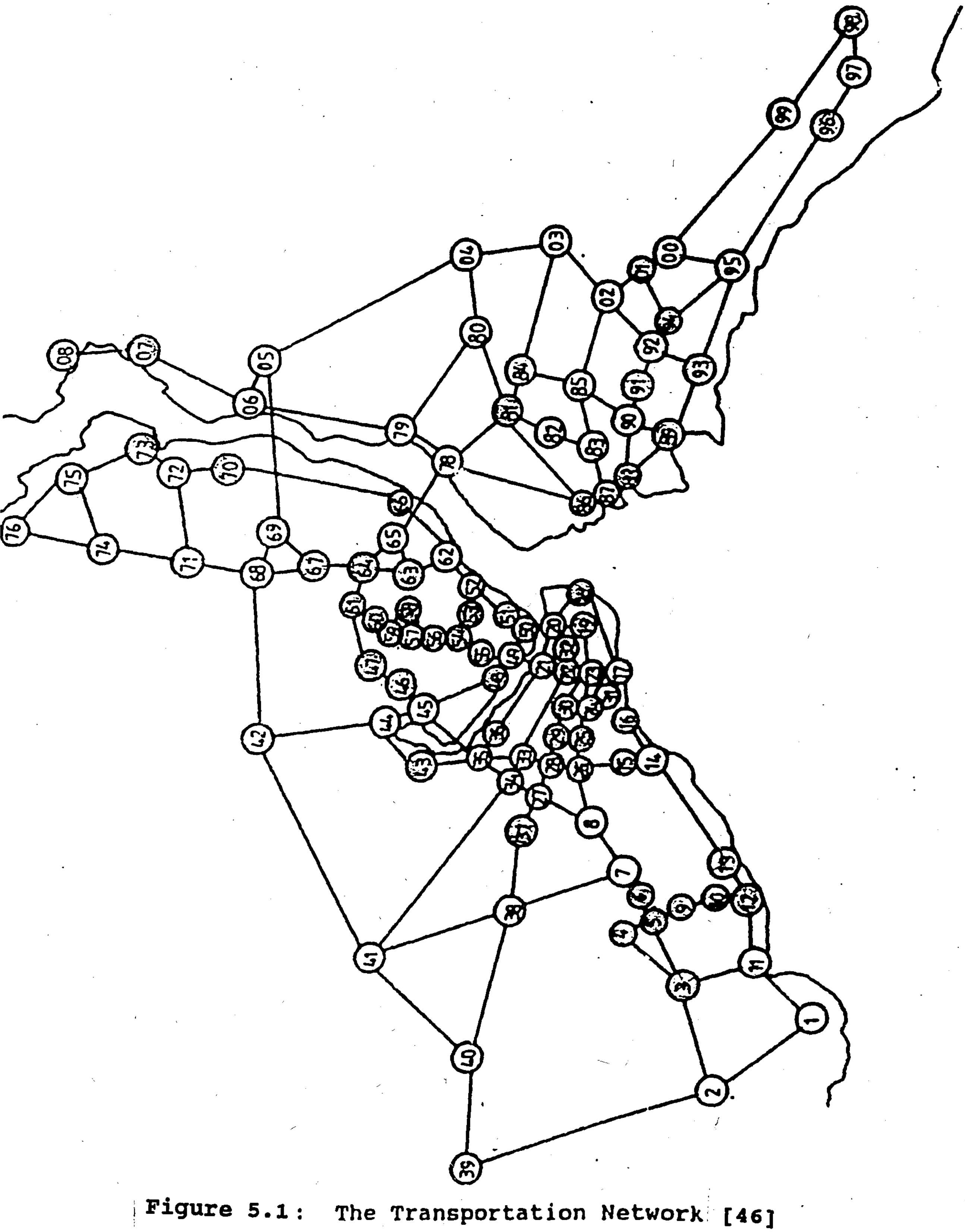


Figure 5.1 : The Transportation Network [46]
(Scale : 1 / 160,000)

[7*(0.160 gr)*(# of beds)] + [2*(# of ops.)/52] * 1.20 =
= [1,12 * (# of beds) + 0,038 * (# of operations)] * 1.20 =
= [1,344 * (# of beds) + 0,046 * (# of operations)]

At the hospitals nodes which have more than one hospital, to determine the amount of wastes, demands of each hospitals represented by that node are added and total amount is considered as the demand at that node.

In this study, two different transportation networks were used. The initial step of the analysis has been the definition of the nodes and arcs of these transportation networks. In the first network nodes are set to correspond to major trafic junctions and/or hospitals and arcs are the connecting major roads. The node and arc descriptions are listed in Appendix-3 and Appendix-4 respectively. The locations of 108 nodes, including 59 hospitals nodes (there can be more than one hospital at a hospital node) and 49 traffic junctions; and the 154 arcs between them are obtained from maps of 1/10.000 and 1/50.000 scale. Details of the transportation network are indicated on Figure 5.1.

The second transportation network (reduced network) was constracted by using the first one. In order to obtain more realistic travel times (and accordingly travel costs) between node pairs, Dijkstra's [48] shortest path algorithm is implemented on the first 108-node network and a cost matrix is created between all nodes. Then only hospitals nodes are pinpointed and the nodes represent the traffic junctions are not considered any further in this new transportation network. Then, this network was partitioned into two smaller networks, namely Reduced Network-1 and Reduced Network-2. The first one represents the Eurepean Part and the second represents the Anatolian side. CAVR algorithm has been implemented on theses reduced networks. Details of the reduced transportation networks are shown on Figure 5.2.

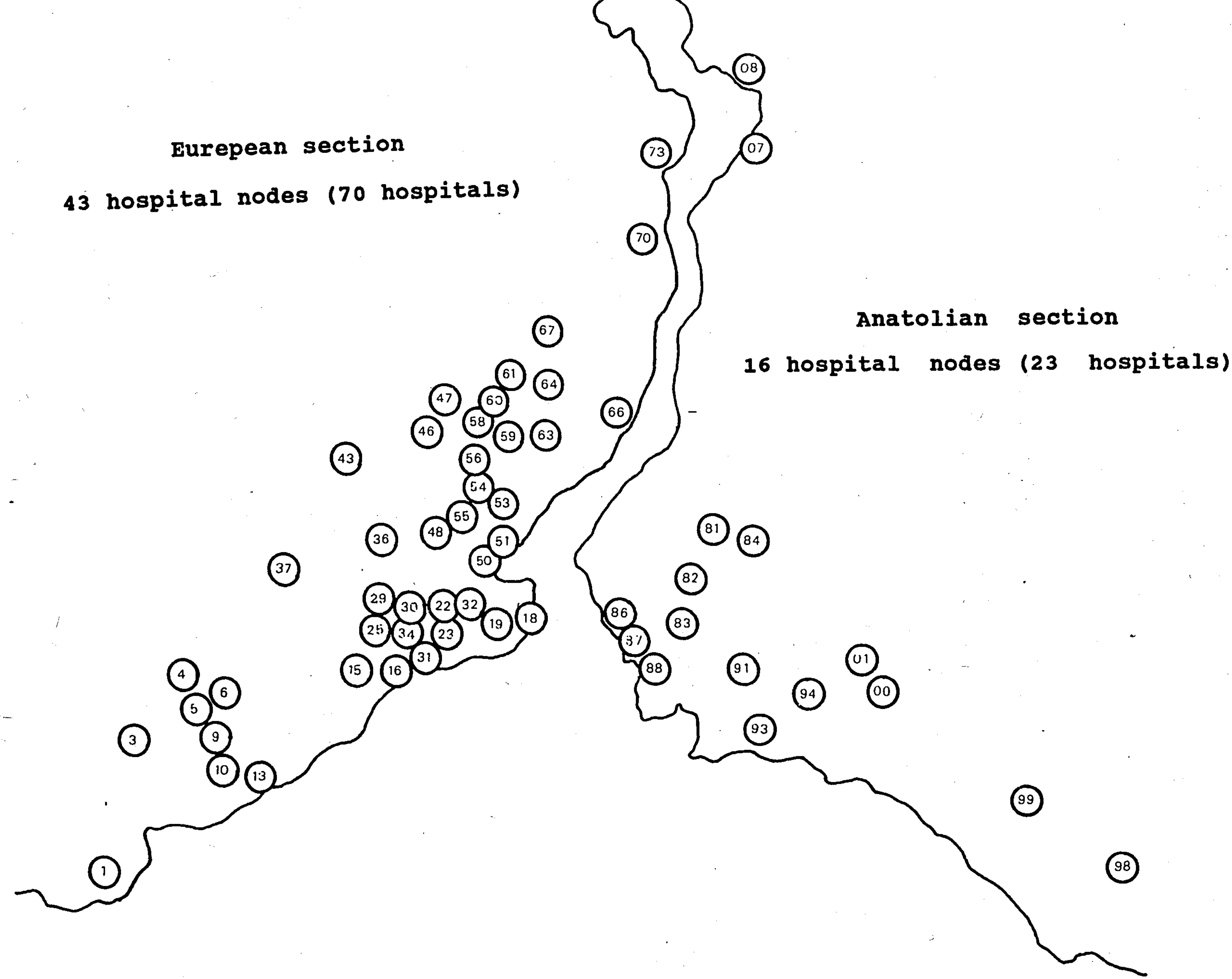


Figure 5.2. Reduced transportation network.

As reflected in these reduced transportation networks, there are 43 hospital nodes (70 hospitals) in the Eurepean section of Istanbul, and 16 hospital nodes (23 hospitals) in the Anatolian section. In addition to these hospital nodes, a disposal site is assumed north-east of the city, near the Black Sea coast, namely in Karakiraz as advised in Akgul's Master Thesis Study [46]. Figure 5.3. shows the selected location as well as the region investigated.

In order to be able to organize city's two sections' collection systems independent each other, it was prefered to divide the transportation network into two parts, and solve two independent and relatively smaller Vehicle Routing Problems. Accordingly, two VRP, one having 44 nodes and the other having 17 nodes including the disposal site, have been solved. In order to see the collections system on the monitor, a coordinate system is set up with its zero point at the Nort-West corner.

5.3. Implementation

The methodology described in the previous section is implemented to determine a good routing design for the vehicles which collect the hazardous solid wastes from the hospitals in Istanbul metropolitan area. The collection operation is executed once in two days because the new government statement requires hazardous wastes to be collected in 48 hours at most. The total amount of hazardous waste generated and collected in Istanbul is around 40 metric tons/week.

In light of this information, the capacity of vehicles used, is varied from 2.000 kg/vehicle to 5.000 kg/vehicle and different routing and collection alternatives are generated.

5.4. Results

The computer code, written in Turbo Pascal, performed quite effectively in an interactive mode. In this case we have been able to obtain various solutions by deleting or changing the existing parameters. Also, the code can give the planner some flexibility in the decision making process.

Four alternative solutions are generated for Eurepean side and three alternative solutions are obtained for Anatolian side with different vehicle capacities values. The results are shown in Table 5.1. and Table 5.2.

Intuitively, assuming Euclidian distance measures, the minimal traveling salesman tour should not intersect itself, a property that is obvious from geometrical considerations. (If there is an intersection or crossover on the tour, an improvement should be possible by eliminating two links and replacing them by two others.) Note that, in some solutions below, intersections can be seen. This is because of the geografical layout of the actual roads and the resulting non-Euclidian distance matrix.

SOLUTION E1.

Vehicle Capacity : 5.000 kg
Number of Routes : 2
Route-1 : (1-41-39-37-26-25-24-22-23-17-16-20-21-14-27-32-30-40-42-43-44-1)
Route-2 : (1-36-35-34-33-31-15-19-18-6-3-2-4-5-7-8-9-10-11-12-13-28-29-38-1)

Route-1 Cost : 149.8 km
Route-2 Cost : 138.0 km

Total Cost : 287.8 km

SOLUTION E2.

Vehicle Capacity : 4.000 kg

Number of Routes : 3

Route-1 : (1-26-25-24-22-21-20-17-16-14-27-32-30-1)

Route-2 : (1-38-31-33-34-35-36-37-39-41-1)

Route-3 : (1-29-28-12-11-10-9-8-7-4-2-3-5-6-23-18-19-15-13-40-42-43-44-1)

Route-1 Cost : 116.8 km

Route-2 Cost : 90.3 km

Route-3 Cost : 162.5 km

Total Cost : 369.6 km

SOLUTION E3.

Vehicle Capacity : 3.000 kg

Number of Routes : 3

Route-1 : (1-12-11-10-9-8-7-4-2-3-5-6-23-18-19-15-13-28-29-38-1)

Route-2 : (1-25-24-22-21-20-17-16-14-27-32-30-1)

Route-3 : (1-41-39-26-37-36-35-34-33-31-40-42-43-44-1)

Route-1 Cost : 140.1 km

Route-2 Cost : 116.8 km

Route-3 Cost : 117.3 km

Total Cost : 374.2 km

SOLUTION E4.

Vehicle Capacity : 2.000 kg

Number of Routes : 5

Route-1 : (1-24-23-6-5-2-3-7-8-9-10-32-1)
Route-2 : (1-13-16-17-18-19-14-38-1)
Route-3 : (1-12-11-15-20-21-22-28-29-1)
Route-4 : (1-30-31-33-34-35-36-37-39-41-1)
Route-5 : (1-26-25-27-40-42-43-44-1)

Route-1 Cost : 140.1 km
Route-2 Cost : 103.6 km
Route-3 Cost : 112.9 km
Route-4 Cost : 90.3 km
Route-5 Cost : 118.1 km

Routing Cost : 565.0 km

Table 5.1. Alternative Solutions for Eurepean Side

SOLUTION A1.
Vehicle Capacity : 4.000 kg
Number of Routes : 1

Route-1 : (1-13-12-14-15-11-10-9-5-2-3-4-6-7-8-17-16-1)
Route-1 Cost : 120.1 km

Total Cost : 120.1 km

SOLUTION A2.
Vehicle Capacity : 3.000 kg
Number of Routes : 2

Route-1 : (1-15-14-11-10-9-5-2-3-4-6-7-8-17-16-1)
Route-2 : (1-13-12-1)

Route-1 Cost : 110.7 km
Route-2 Cost : 76.0 km

Total Cost : 186.7 km

SOLUTION A3.

Vehicle Capacity : 2.000 kg
Number of Routes : 2

Route-1 : (1-11-10-9-8-7-6-4-3-2-1)
Route-2 : (1-17-16-5-15-14-13-12-1)

Route-1 Cost : 88.7 km
Route-2 Cost : 106.7 km

Total Cost : 195.4 km

Table 5.2. Alternative Solutions for Anatolian Side

Note that, in all these solutions routes are given on the reduced networks. So, in order to obtain the real routes , an additional tracing operation is required. Given a cycle occurring in the solution , one can replace each arc (i-j) in the reduced network with the arcs which are builts the shortest path between i and j in the original transportation network. After this transformation , detailed real solutions can be applied on the city's transpotation map.

As an example, consider the Solution-3 for Anatolian side. In the solution-3, there are two routes as follows :

Route-1 : (1-11-10-9-8-7-6-4-3-2-1)
Route-2 : (1-17-16-5-15-14-13-12-1)

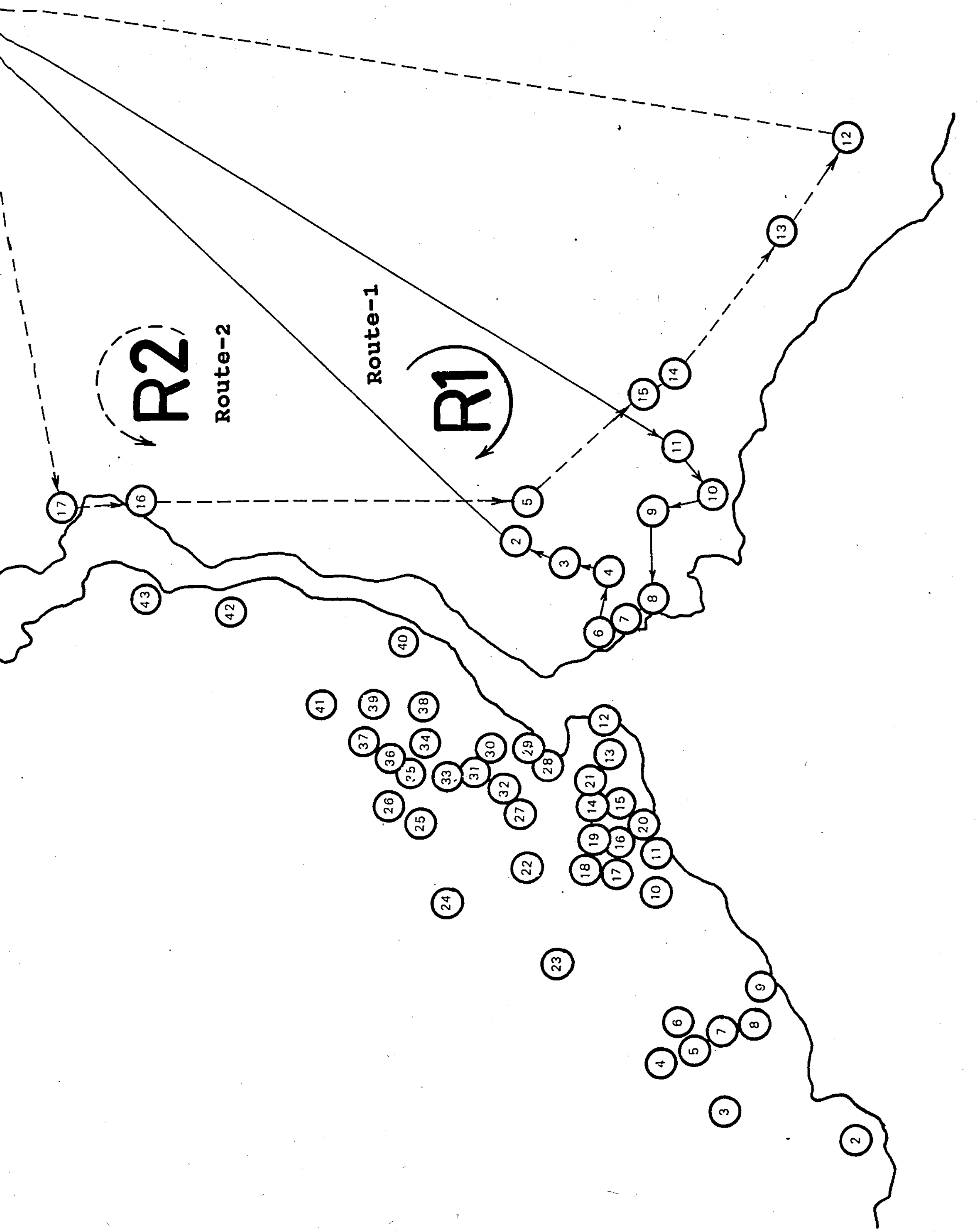


Figure 5.3. Final routes on reduced network

In this solution, nodes and routes are given on the reduced network as can be seen on Figure 5.3. Nodes can easily be transfered according to the original network by refering to a cross reference table such as Table 5.3.

Node # on Reduced Network-2	Node # on Original Network
1	Non-exist (depot)
2	81
3	82
4	83
5	84
6	86
7	87
8	88
9	91
10	93
11	94
12	98
13	99
14	00
15	01
16	07
17	08

Table 5.3. **Node information**

Solution-7 can be rewritten in terms of original node numbers as follows :

Route-1 : (1-94-93-91-88-87-86-83-82-81-1)
Route-2 : (1-08-07-84-01-00-99-98-1)

In order to transfer the routes to the original network, the shortest paths between the nodes in these routes, have to be traced and the detailed routes will then be available on the original transportation network.

As an example, consider the Route-1, in order to reach to node-94 from node-1, the path { 1-04-03-02-01-94 } must be followed. Similarly, a vehicle has to pass through node-92 in order to collect the waste produced at node-93. So, all paths can be determined in this way and the final routes can be constracted on the original transportation network as shown in Figure 5.4 .

The final routes are :

Route-1 : (**1**-04-03-02-01-**94**-92-**93**-92-**91**-90-**88**-**87**-**86**-87-**83**-**82**-**81**-**1**)

Route-2 : (**1**-**08**-**07**-06-79-78-81-**84**-85-02-**01**-**00**-**99**-**98**-**1**)

Execution times are at most 26 seconds, for these alternative solutions. Each vehicle type has a fixed investment cost and a variable routing cost that is proportional to distance travelled. The variable cost component consists of the cost of fuel, maintenance and manpower. Note that , a vehicle type can be chosen according to the four criteria : availability of vehicle, investment cost, operating cost and compatibility of vehicle size with all conditions . A small vehicle can be more adequate than a big truck, especially in regions with narrow roads. Also, the investment and operating costs are more reasonable for those small vehicles. On the other hand, the total distance travelled from disposal site to collection region increases with number of vehicles used. Since going back and forth between disposal site and collection area causes a growth in total cost.

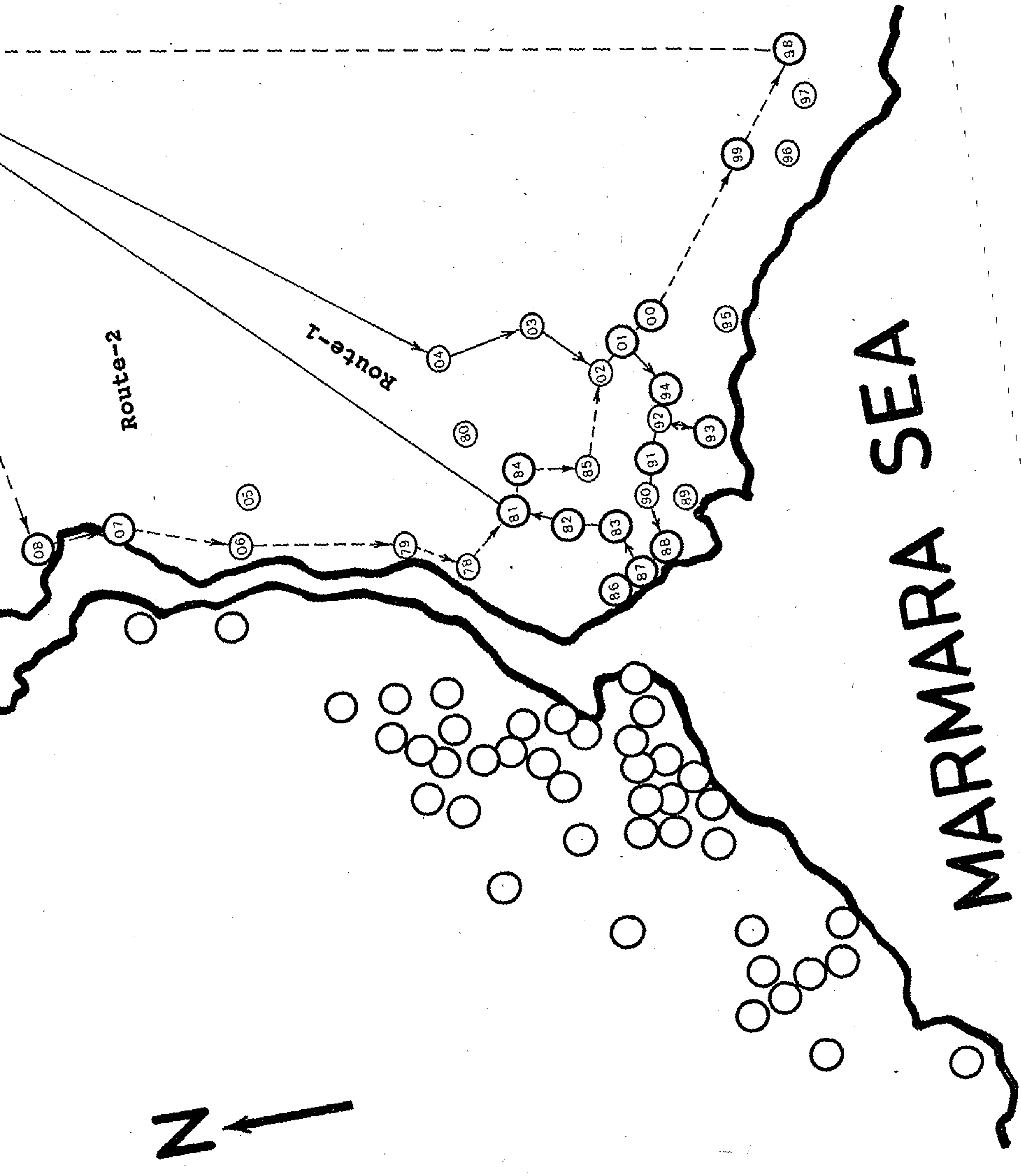


Figure 5.4. Detailed real solutions.

5.5. Several Scenarios

By using these solutions and considering the other aspects mentioned above, proper collection strategy can be created for Istanbul Hospitals.

Cost of collection is assumed to be a function of the collection frequency, the speed of of collection, the quantity of wastes and the work rules.

Since the collection frequency is "once in two days", one collection is of two days of hazardous solid waste accumulation. The speed of collection within area is a function of the quantity of the solid waste produced at hospitals. We assumed 10 minutes per hospital is enough to load the solid waste to a vehicle. The speed in traffic of a collection vehicle is assumed as 60 km/per hour for both loaded and unloaded vehicles. For large vehicles one driver and two workers can be used and for small vehicles a driver and a worker can be employed.

SCENARIO 1.

Vehicle Capacity : 2.000 kg
Number of Vehicles : 2
Total Distance : 760.4 km
Total Time : 23.67 hours

ROUTE	DISTANCE	# OF NODES	TIME
E1	140.1 km	13	4.50 hours
E2	103.6 km	8	3.06 hours
E3	112.9 km	9	3.38 hours
E4	90.3 km	10	3.17 hours
E5	118.1 km	8	3.30 hours
A1	88.7 km	10	3.15 hours
A2	106.7 km	8	3.11 hours

These routes might be grouped so that a vehicle can serve them in the same day.

GROUPED ROUTES	TIME	VEHICLE NO.	SERVICE DAYS
E1	4.50 hours	1	Mon, Wed, Fri, Sun
E2 + E3	6.44 hours	2	
E4 + E5	6.47 hours	1	Tue, Thu, Sat
A1 + A2	6.26 hours	2	

SCENARIO 2.

Vehicle Capacity : 3.000 kg
Number of Vehicles : 2
Total Distance : 560.9 km
Total Time : 20.03 hours

ROUTE	DISTANCE	# OF NODES	TIME
E1	140.1 km	20	5.67 hours
E2	116.8 km	12	3.95 hours
E3	117.3 km	14	4.29 hours
A1	110.7 km	15	4.35 hours
A2	76.0 km	3	1.77 hours

These routes might be grouped so that a vehicle can serve them in the same day.

GROUPED ROUTES	TIME	VEHICLE NO.	SERVICE DAYS
E1	5.67 hours	1	Mon, Wed, Fri, Sun
E2 + E3	8.24 hours	2	
A1 + A2	6.12 hours	1 or 2	Tue, Thu, Sat

SCENARIO 3.

Vehicle Capacity : 4.000 kg
Number of Vehicles : 1
Total Distance : 489.7 km
Total Time : 18.65 hours

ROUTE	DISTANCE	# OF NODES	TIME
E1	116.8 km	13	4.11 hours
E2	90.3 km	10	3.17 hours
E3	162.5 km	23	6.54 hours
A1	120.1 km	17	4.83 hours

These routes might be grouped so that a vehicle can serve them in the same day.

GROUPED ROUTES	TIME	VEHICLE NO.	SERVICE DAYS
E1 + A1	8.94 hours	1	Mon, Wed, Fri, Sun
E2 + E3	9.71 hours	1	Tue, Thu, Sat

SCENARIO 4.

Vehicle Capacity : 5.000 kg
Number of Vehicles : 1
Total Distance : 407.9 km
Total Time : 17.12 hours

ROUTE	DISTANCE	# OF NODES	TIME
E1	149.8 km	22	6.16 hours
E2	138.0 km	23	6.13 hours
A1	120.1 km	17	4.83 hours

These routes might be grouped so that a vehicle can serve them in the same day.

GROUPED ROUTES	TIME	VEHICLE NO.	SERVICE DAYS
E1 + A1	10.99 hours	1	Mon, Wed, Fri, Sun
E2	6.13 hours	1	Tue, Thu, Sat

To summarize the results of these scenarios Table 5.4. can be used.

CAPACITY	2000	3000	4000	5000
NUMBER OF VEHICLES	2	2	1	1
TOTAL DISTANCE	760.4 km	560.9 km	489.7 km	407.9 km
TOTAL TIME	23.67 hr	20.03 hr	18.65 hr	17.12 hr
NUMBER OF TEAM	2	one day 1 one day 2	1	1
WORKING CONDITIONS	everyday normal	everyday normal	everyday overtime	one day normal one day overtime

Table 5.4. Alternative Scenerios.

VI. CONCLUCIONS

The Vehicle Routing Problem (VRP) is an important and practical problem which has received considerable attention in the literature as summarized in section 2. In this study several general solution techniques for this problem have been discussed and a new algorithm, CAVRS (Computer Assisted Vehicle Routing System) has been developed to solve the standart VRP.

The performance of CAVRS has been compared with several methods in the literature and the outcome have been quite satisfactory, as explained in section 4. As far as execution times and objective function values (of some standart test problems) are concerned CAVRS is very much competitive with some of the best known algorithms in the literature. The algorithm is also easy to ipmlement and use, and does not require high computational capability.

Furthermore with CAVRS a user can easily input data and find a solution for a given VRP, and then analyze this solution step by step, by using the rich graphic interface provided by the system.

Also, this implementation has also shown that CAVRS is a system having a large variety of potential applications.

On the other hand, this study has shown that some complex municipal solid waste collection problems can be handled quite effectively through vehicle routing modelling and solution procedures. The implementation of CAVRS to the hazardous waste collection problem of Istanbul, as explained in section 5, has generated some promising results. Of course, the data used for this implementation has mainly been preliminary data with unconfirmed reliabilty. Another shortcoming of this implementation is the singular emphasis on transportation costs. Whereas in reality other aspects of transporting hazardous material, such as accidental spilling risks, have to be considered. But, is the least, the applicability of the approach has been demonstrated and need for reliable data in this regard (hazardous waste generations costs, disposal sites, available transporation routes, accident probabilities) has been highlited.

BIBLIOGRAPY

1. Marks, D., J.Cohan, H. Moore and R. Stricker, "Routing for Municipal Services", Proc. of the 6th Annual ACM Urban Symposium, 1971.

2. Held, M., R. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, No. 6, pp. 1138-1162.

3. Held, M., R. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees II", Mathematical Programming, Vol. 1, pp. 1138-1162, 1971.

4. Lin, S., "Computer Solutions of the TSP", Bell System Technical Journal 44, pp. 2245-2269, 1965.

5. Little, J., K. Murty, D. Sweeney and C. Karel, "An Algorithm for the TSP", Operations Research, Vol. 11, No. 6, pp. 972-989, 1963.

6. Liebman, J., "Notes on Edmonds' Matching and White' k-Matching", Unpublished Class Notes, June 1972.

7. Liebling, T. "Graphen Theroie in Plaunungs- und Touren Problemen", Lecture nodes in Operation Research, Vol.21, Springer-Verlag 1970.

8. Edmonds, J., E.Johnson, "Matching, Euler Tours and the Chineese Postman Problem", Math. Prog., 5, 88-124, 1973.

9. Christofides, N., "The Optimal Traversal of a Graph.", Report 71/16 of the Imperial College of Science, November 1971.

10. Wen-Lea Pearn, Arjang Assad and Bruce L. Golden, "Transforming Arc Routing Into Node Routing Problems.", Comput. Opns. Res., Vol. 14, No. 4, pp. 285-288, 1987.

11. Rosenkrantz,R., R. Sterns and P. Lewis, "An Analysis of Several Heuristics for the Traveling Salesman Problem.", SIAM J, Comp., Vol. 6, pp. 563-581, 1977.

12. Clarke, G. and J.W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,", Ops. Res., Vol. 12, pp. 568-581, 1964.

13. Golden, B., "A Statistical Approach to the TSP.", Networks, Vol. 7, pp. 209-225, 1977.

14. Lin, S., "Computer Solutions to the Traveling Salesman Problem.", Bell System Technical Journal, 44, 2245-2269, 1965.

15. Lin, S., and B.W. Kerninghan, "An Effective Heuristic Algorithm for the TSP.", Operations Research, 21, 498-516, 1973.

16. Lavyer, E.L., S.K. Lenstra, A.H.G. Rinnoci Kan and D.B. Shmays, "The Traveling Salesman Problem.", pp. 218-219, 1986.

17. Golden, B.L.,W. Stewart, "The Emprical Analysis of TSP Heuristics.", pp. 30-35, 1983.

18. Golden, B., T. Magnanti and H. Nguyen, "Implementing Vehicle Routing Algorithms.", Networks, Vol. 7, pp. 113-148, 1977.

19. Gillett, B.E., and L.R. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem.", Operations Research, pp. 340-349, 1976.

20. Gillett, B.E., and J.Johnson , "Sweep Algorithm for the Multiple Depot Vehicle Dispatch Problem.", ORSA National Meeting, San Juan, Fall 1974.

21. Karp, R.,"Probabilistic Analysis of Partitioning Algorithms for the TSP in the Plane,", Math. Operations Research, Vol. 2, pp. 209-224, 1977.

22. Golden, B., A. Assad, L. Levy, and F. Gheysens, "The Fleet Size and Mix Vehicle Routing Problem.", Management Science & Statistics Working Paper No.82-020, University of Maryland at College Park, 1982.

23. Newton R. and W. Thomas, "Design of School Bus Routes by Computer.", Socio -Economic Planning Sci., 3, pp. 75-85, 1969.

24. Bodin L. and L. Berman, "Routing and Scheduling of School Buses by computer.", Transportation Sci., Vol. 13, pp. 113-129, 1979.

25. Bodin L., S. Kursh, "A Computer Assisted System for the Routing and Scheduling of Sstreet Sweepers.", Ops. Res., Vol. 26, pp. 525-537, 1978.

26. Stern, H. and M. Dror, "Routing Electric Meter Readers.", Comput. Ops. Res., Vol. 6, pp. 209-223, 1979.

27. Golden, B. and R. Wong, "Capacitated Arc Routing Problems.", Networks, Vol. 11, pp. 305-315, 1981.

28. Christofides, N. and S. Eilon, "Algorithms for Large Scale Traveling Salesman Problems.", Operational Research Quarterly.", Vol. 23, pp. 511-518, 1972.

29. Russel, R., "An Effective Heuristic for the M_Tour Traveling Salesman Problem with Some Side Constraints.", Ope. Res., Vol. 25, pp. 517-524, 1977.

30. Held, M. and R. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees.", Ops. Res., Vol. 18, pp. 1138-1162, 1970.

31. Hansen, K. and J. Krarup, "Improvements oh the Held-Karp Algorithm for the Symmetric Traveling Salesman Problem.", Math. Prog., Vol. 7, pp. 87-96, 1974.

32. Balas, E. and N. Chiristofides, "A Restricted Lagrengian Approach to the Traveling Salesman Problem.", Math. Prog., Vol. 21, pp. 19-46, 1981.

33. Crowder, H. and M. Padberg, "Solving Large Scale Symmetric Traveling Salesman Problems to Optimality.", Management Sci., Vol. 26, pp. 495-509, 1980.

34. Bodin, L., G. Fagan, R. Welenby and J. Greenberg, "The Design of a Computarized Sanitation Vehicle Routing and Scheduling System for the Town of Oyster Bay, New York.", Comput. Ops. Res., Vol. 16, pp. 45-54, 1989.

35. Golden, B.L., L. Bodin and T. Godwin, "Microcomputer Based Vehicle Routing and Scheduling Software.", Comput. Ops. Res., Vol. 13, pp. 277-285, 1986.

36. Fischer, M., A. Greenfield, R. Jaikumar and J. Lester, "A Computerized Vehicle Routing Application.", Interfaces, Vol. 12, pp. 42-52, 1982.

37. Bell, W., L. Dalberto, M. Fischer, A. Greenfield, R. Jaikumar, R. Mack and P. Prutzman, "Improving the Distribution of Industrial Gases with an On-Line Computarized

Routing and Scheduling System.', _Interfaces,_ Vol. 13, pp. 4-23, 1983.

38. Brown, G. and G. Graves, "Real tTime Dispatch of Petrolium Tanks Trucks.", _Management Sci._, Vol. 27, pp. 19-31, 1981.

39. Evans, S. and J. Norback, "The Impact of a Decision Support System for Vehicle Routing in a Food Service Supply Sitiation.", _Jou. Ops. Res. Soc._, Vol 36, pp. 467472, 1985.

40. Belardo, S., P. Duchessi and J. Seagle, "Microcomputer Graphics in Support of Vehicle Fleet Routing.", _Interfaces,_ Vol. 15, pp. 84-92, 1985.

41. Jean-Yves Potvin, Guy Lapalme and Jean-Marc Rousseau, " Alto : A Computer System for the Design of Vehicle Routing Algorithms.", _Comput. Ops. Res._, Vol. 16, pp. 451-470, 1989.

42. Jean-Yves Potvin, Guy Lapalme and Jean-Marc Rousseau, "Integration of AI and OR Techniques for Computer-Aided Algorithmic Design in the Vehicle Routing Domain.", _Jou. Ops. Res. Soc._, Vol. 41, pp. 517-525, 1990.

43. Croes, A., "A Method for Solving Traveling Salesman Problems.", _Ops. Res._, Vol. 5, pp. 791-812, 1958.

44. Dror, M. and L. Larry, "A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementtation for Inventory Routing.", _Comput. Ops. Res._, Vol. 13, pp. 33-45, 1985.

45. Paessens, H., " The Savings Algorithm for the Vehicle Routing Problem.", _European Journal of Ops. Res._, Vol. 34, pp. 336-344, 1988.

46. Akgül, M., "An Optimization Approach to the Transportation of Hazardous Solid Waste and the Location of Disposal Facilities.", MSc. Thesis, Departmant of Industrial Engineering, Boğaziçi University, 1991.

47. Yataklı Tedavi Kurumlari İstatistik Yıllığı 1991 , Sağlik Bakanlığı Yayınları,1991.

APPENDIX

APP-1 : Hospital Nodes for Eurepean Part

APP-2 : Hospital Nodes for Anatolian Part

APP-3 : Arc Descriptions

APP-4 : Node Descriptions

APP-5 : Computer Screens of CAVR

APPENDIX 1. Node Descriptions for Anatolian Side.
(1 depot, 16 nodes and 23 hospitals.)

Node No	Hospitals in the Node	# beds/ # operations
1	Disposal Site in Karakiraz	Depot
2	Marmara Üniversitesi Hastanesi	250 / 2072
3	Polis Hastanesi Validebağ Provantoryum-Sanatoryum Validebağ Öğretmen Hastanesi	504 / 1274
4	Koşuyolu Göğüs Hastanesi	115 / 841
5	Çamlıca G. Hasta. Askeri Hast.	57 / 100
6	Zeynep Kamil Hastanesi	553 / 5328
7	Haydarpaşa Numune Hastanesi Haydarpaşa G. K. D. Cer. Merkezi Gülhane Askeri Tıp Akademisi Hast.	952 / 14806
8	Kadıköy Sağlık Merkezi	30 / 974
9	Göztepe SSK Hastanesi	1104 / 12138

10	Hayvan Hastanesi	*
11	SSK Geriatri Hastanesi SSK Psikiatri Hastanesi	246 / 0
12	Kartal Devlet Hastanesi	287 / 1700
13	SSK Süreyyapaşa Sanatoryumu SSK Provantoryum ve Rehabilitasyon Mer.	1800 / 2393
14	Organ Nakli Hastanesi	30 / 100
15	PTT Hastanesi	396 / 1706
16	Beykoz Hastanesi Paşabahçe Hastanesi	212 / 7718
17	Beykoz Çocuk Göğüs Hast. Hastanesi	123 / 0
	TOTAL :	7.933 / 50.920

* It is assumed that , the total solid waste produced in this hospital is nearly 45 kg/week.

APPENDIX 2. Node Descriptions for Eurepean Side. (1 depot, 43 nodes and 70 hospitals.)

Node No	Hospitals in the Node	# beds/# operations
1	Disposal Site in Karakiraz	Depot
2	International Hospital	130 / 400
3	Rehabilitasyon Merkezi	130 / 0
4	Yaşam Hastenesi	82 / 53
5	Özel İncirli Hastanesi	49 / 369
6	Ahmet Ermis Hastanesi	30 / 100
7	Bakırköy Ruh ve Sinir Hast. Hastanesi Bakırköy Devlet Hastanesi İstanbul Lepra Hastanesi	2369 / 1772
8	Özel Omur Hastanesi SSK Doğumevi	532 / 6062

9	Sümerbank Merkez Hastanesi	64 / 241
10	Yedikule Göğüs Hast. Hastanesi Balıklı Rum Hastanesi Ermeni Hastanesi	960 / 1867
11	SSK İstanbul Hastanesi	804 / 11501
12	Deri ve Tenasul Hast. Hastanesi	70 / 0
13	Kuduz Hastanesi	30 / 0
14	Özel Fatih Hastanesi Hifsizsihha Enstit.	5 / 140
15	Doğaner Hastanesi	30 / 259
16	Haseki Hastanesi	492 / 2342
17	Çapa Tıp Fakültesi Hastanesi Çapa Özben Hastanesi	2468 / 15408
18	Vakıf Gureba Hastanesi Topkapı Hastanesi	484 / 3139
19	Vatan Hastanesi	163 / 400

20	Cerrahpaşa Tıp Fakültesi	1640 / 10903
21	Süleymaniye Doğumevi Esnaf Hastanesi	310 / 2721
22	Musevi Hastanesi	80 / 62
23	Sağmalcılar Devlet Hastanesi Özel Bayrampaşa Merkez Hastanesi	129 / 247
24	SSK Eyüp Hastanesi	209 / 1709
25	Darulaceze Hastanesi	555 / 8
26	SSK Okmeydanı Hastanesi Bulgar Hastanesi	1074 / 10209
27	Kasımpaşa Deniz Hastanesi	100 / 400
28	Beyoğlu Hastanesi St. George Hastanesi	258 / 3527
29	Denizcilik Bankası Hastanesi İtalyan Hastanesi	125 / 579
30	Gümüşsuyu Askeri Hastanesi	100 / 400

31	Taksim Hastanesi Alman Hastanesi	398 / 5028
32	Tepebaşı Vatan Hastanesi	53 / 200
33	Fransız Pastör Hastanesi Ermeni Katolik Surp Agop Hastanesi	100 / 1794
34	Teşvikiye Sağlik Evi SSK Kadın Doğum Kliniği Amerikan Bristol Hastanesi Güzelbahçe Kliniği Pakize Tarzi Kliniği	309 / 4723
35	Hayvan Hastanesi	*
36	Şişli Eftal Hastanesi SSK Şişli Hastanesi Ataman Kliniği Can Hastanesi Türk Kalp Vakfı Osmanoğlu Kliniği Fransız La Paix Hastanesi Hayat Hastanesi	1135 / 25249

37	Aksoy Hastanesi Mecidiyekoy Çevre Hastanesi	70 / 1358
38	Sait Çiftci Kamu Sağlığı Merkezi	30 / 100
39	TKV Onkoloji Merkezi	30 / 100
40	Ortaköy Şifa Yurdu	20 / 80
41	Özel Levent Hastanesi	12 / 96
42	Baltalimanı Kemik Hast. Hastanesi	236 / 1675
43	İstinye Devlet Hastanesi	200 / 1257
44	İsmail Akgün Devlet Hastanesi	25 / 193
	TOTAL :	16.165 / 103.347

* It is assumed that , the total solid waste produced in this hospital is nearly 45 kg/week.

APPENDIX 3: Arc Descriptions

Start-End Node	Description	Arc Length L(km)
1-2	Atatürk Hava Limanı Yolu	4.20
1-11	Rauf Orbay Cad.	2.20
2-3	E-5 Ataköy	2.80
2-39	TOYBY-Sefaköy	7.60
3-4	Eski Londra Asfaltı	1.30
	Bahçelievler Cad.	0.60
3-5	E-5 Bakırköy	1.80
3-11	Kilitbahir Cad.	2.00
4-5	İzzettin Çalislar Cad.	1.10
5-6	E-5 Osmaniye	1.30
5-9	Bağlarbaşı-İncirli Cad.	0.50
6-7	E-5 Merter	0.90
7-8	E-5 Tozkoparan	2.20
7-38	TOYBY-Bayrampaşa	5.40
8-26	Londra Asfalti	1.00
8-27	E-5 Topkapı	1.60
9-10	Bağlarbaşı-İncirli Cad.	1.30
10-12	Ebuzziya Cad.	0.50
	Bakırköy İstasyon Cad.	0.30
12-13	Kennedy Cad.	1.20
13-14	Kennedy Cad.	3.70
14-15	Genç Osman Cad.	1.00
14-16	Kennedy Cad.	2.20
15-26	Mevlanakapı-Topkapi Yolu	0.70
	10.Yil Cad.	0.90
	Belgradkapi-Demirhane Yolu	0.90
16-17	Kennedy Cad.	1.80
17-18	Kennedy Cad.	3.00
17-23	M.Kemal Cad.	0.70
18-20	Kennedy Cad.	2.40
19-20	Ankara Cad.	0.70
	Alemdar Cad.	0.40
19-23	Ordu Cad.	1.00
	Yeniçeriler Cad.	0.50
	Divanyolu Cad.	0.50
20-21	Sobacılar Cad.	0.40
	Ragip Gümüşpala Cad.	0.70
20-50	Galata Köprüsü	0.70
	Karaköy Cad.	0.30
21-22	Atatürk Bulvarı	1.20
21-36	Abdulezelpaşa Cad.	1.60
	Demirhisar Cad.	0.60
21-49	Atatürk Köprüsü	0.70
22-23	Atatürk Bulvari	0.60
22-32	Şehzadebaşı Cad.	1.00
22-33	Fevzipaşa Cad.	1.90
	Macar Kardeşler Cad.	0.60
23-24	Millet Cad.	0.70
23-30	Vatan Cad.	0.70
23-31	Cerrahpaşa Cad.	1.00

24-25	Millet Cad.	1.00
25-26	Millet Cad.	0.80
26-28	Topkapı-Edirnekapı Cad.	0.70
27-28	BKBY-Edirnekapı	0.60
27-34	E-5 Edirnekapı	0.80
27-37	TOYBY-Sağmalcılar	1.60
28-29	Vatan Cad.	0.80
28-33	Topkapı-Edirnekapı Cad.	0.90
29-30	Vatan Cad.	1.00
33-34	Rami Kışla Cad	0.60
33-35	Savaklar Cad.	1.20
34-35	BKCY-Ayvansaray	1.20
34-41	Rami Kışla Cad	3.00
	İstanbul-Edirne Cad.	4.50
35-36	Ayvansaray Cad.	0.80
35-43	Eyüp Kazikli Yol	1.20
	Silahtarağa Cad.	1.20
35-45	Fatih Köprüsü	0.70
	BKCY-Haliç	1.80
37-38	TOYBY-Esenler	2.50
38-40	TOYBY-Bağcılar	4.20
38-41	TOYBY-Hal	4.20
39-40	TOY-Mahmutbey	3.50
40-41	TOY-Atışalani	3.80
41-42	TOY-G.O.Paşa	8.00
42-44	FSMFCY-Kağıthane	3.80
42-68	FSMKCY-Sanayi	4.40
43-44	Silahtarağa Cad.	2.50
	Sünnet Köprüsü	1.00
44-45	FSMKCY-Talatpaşa	1.80
45-46	BKCY-Darülaceze	0.90
45-48	Fatih Sultan Mimberi Cad.	2.70
46-47	BKCY-Okmeydanı	1.20
47-61	BKCY-Şişli	2.30
48-49	Evliya Çelebi Cad.	1.20
49-50	Tersane Cad.	0.70
49-55	Yolcuzade İskender Cad.	0.50
	Meşrutiyet Cad.	1.00
50-51	Kemeralti Cad.	0.80
51-52	Necatibey Cad.	1.00
	Meclisi Mebusan Cad.	0.80
52-53	Kadirgalar Cad.	0.30
	İnönü Cad.	0.60
52-62	Beşiktaş Cad.	0.40
	Dolmabahçe Cad.	0.60
53-54	İnönü Cad.	0.70
54-55	Tarlabaşi Bulvarı	0.80
54-56	Cumhuriyet Cad.	0.50
56-57	Cumhuriyet Cad.	0.60
57-58	Vali Konağı Cad.	0.70
57-59	Halaskargazi Cad.	0.50
58-59	Rumeli Cad.	0.50
59-60	Halaskargazi Cad.	0.80
60-61	Büyükdere Cad.	2.00
	Şişli Yıldız-Posta Cad.	0.60
61-64	BKCY-Esentepe	0.70

62-63	Barbaros Bulvarı	1.00
62-66	Muallim Naci Cad.	0.70
	Cırağan Cad.	1.60
63-64	Barbaros Bulvarı	1.70
63-65	Beşiktaş-BKBY	1.00
64-65	BKCY-Balmumcu	1.10
64-67	Büyükdere Cad.	2.00
65-78	BKCY-Ortaköy	1.20
	Boğaz Köprüsü	1.20
66-70	Muallim Naci Cad.	1.20
	Sahil Yolu	1.40
	Yahya Kemal Beyatlı Cad.	1.00
	Rumeli Hisarı Cad.	1.80
	Baltalimanı Cad.	0.60
67-68	Büyükdere Cad.	2.00
67-69	Levent-FSMKBY	1.80
68-69	FSMKCY-Harp Akademileri	1.80
68-71	Büyükdere Cad.	1.70
69-105	FSMKCY-Boğaziçi	4.80
70-72	Boyaçiköy-Emirgan Cad.	0.80
	Mirgun İstinye Cad.	1.10
71-72	Maslak Yolu	2.90
71-74	Büyükdere Cad.	3.70
72-73	İstinye Cad.	0.60
73-75	Tarabya-Yeniköy Cad.	2.40
	Köybaşı Cad.	1.60
74-75	Tarabya Cad.	2.30
74-76	Haci Osman Bayırı	1.90
75-76	Kireçburnu Cad.	2.80
76-77	Kazıkli Yol	1.40
	Piyasa Cad.	0.60
	Mesarburnu Cad.	0.80
78-79	Yalıboyu Cad.	2.00
78-81	BKCY-Beylerbeyi	2.00
78-86	Paşalimanı Cad.	2.60
	Hakimiyet-i Milliye Cad.	0.60
	Gündoğumu Cad.	0.80
79-80	Nato Yolu Cad.	5.00
79-106	Kanlıca Anadolu Hisarı Cad.	1.20
	Kandilli Cad.	3.30
	Çengelköy Kuleli Cad.	1.50
80-81	Kısıklı Cad.	3.20
80-104	Alemdağ Cad.	3.10
81-82	Tophanelioğlu Cad.	1.00
81-84	BKCY-Altunizade	1.90
81-86	Nuhkuyusu cad.	2.70
82-83	Koşuyolu Cad.	1.20
83-85	E-5 Koşuyolu	2.20
83-87	İstanbul Ankara Yolu	0.60
84-85	BKCY-Acibadem	1.30
84-103	AOY-Esatpaşa	5.30
85-90	BKCY-Hasanpaşa	1.70
85-102	E-5 Göztepe	3.70
86-87	Eyüp Aksoy Cad.	1.00
87-88	Tıbbiye Cad.	0.50
	Rıhtım Cad.	1.50

88-89	Recep Peker Cad.	1.20
88-90	Söğütlüçeşme Cad.	0.90
	Kayışdağı Cad.	0.50
89-90	Bağdat Cad.	1.00
89-93	Bağdat Cad.	3.30
90-91	Fahrettin Kerim Gökay Cad.	1.60
91-92	Fahrettin Kerim Gökay Cad.	1.70
92-93	Ethem Efendi Cad.	1.70
92-94	Şemsettin Günaltay Cad.	1.30
92-102	Yeldeğirmeni Cad.	1.50
93-95	Bağdat Cad.	3.20
94-95	Şemşettin Günaltay Cad.	2.00
94-101	Sahrayi Cedid-İçerenköy Cad.	1.40
95-96	Bağdat Cad.	4.50
95-100	Ali Nihat Tarlan Cad.	1.40
96-97	Bağdat Cad.	2.00
96-99	Atatürk Cad.	1.40
97-98	Tugayyolu Cad.	1.90
98-99	E-5 Maltepe	3.70
99-100	E-5 Bostancı	4.20
100-101	E-5 İcerenköy	0.70
101-102	E-5 Kozyatağı	1.40
102-103	FSMKCY-Küçükbakkal	2.30
103-104	FSMKCY-Çakmak	3.10
104-105	FSMKCY-Ümraniye	8.80
105-106	Kavacık Çiftliği Cad.	2.00
106-107	Paşabahçe Cubuklu Cad.	1.30
	Kanlıca Çubuklu Cad.	2.20
107-108	Beykoz İncirliköy Cad.	1.80
	Fevzipaşa Cad.	1.50

APPENDIX 4. Node Descriptions

Node No	Description	Hospitals in the Node
1	Atatürk Hava Limani Y. / Havalimanı Yeşilköy Y.	Haifawi Tıp Merkezi
2	Çobançeşme K. K.	
3	Ataköy K. K.	Rehabilitasyon Merkezi
4	Bahçelievler C. / İzzettin Calıslar C.	Yaşam H.
5	İncirli K. K.	Özel İncirli H.
6	E-5 Osmaniye	Ahmet Ermis H.
7	Merter K. K.	
8	E-5 Tozkoparan / Londra Asfaltı	
9	Bağlarbası İncirli C. / Tevfik Sağlam C.	Bakırköy Ruh ve Sinir H. H.
		Bakırköy Devlet H.
		İstanbul Lepra H.
10	Bakırköy İstasyon C. / Bağlarbaşı İncirli C.	Özel Omur H.
		SSK Doğumevi
11	Kilitbahir C. / Rauf Orbay C.	
12	Kennedy C. / Ebuzziya C.	
13	Kennedy C. / İstanbul C.	Sümerbank Merkez H.
14	Kennedy C. / Genç Osman C.	
15	Belgrat Kapısı - Demirhane Y.	Yedikule Göğüs H. H.
		Balıklı Rum H.
		Ermeni H.
16	Kennedy C.	SSK İstanbul H.
17	Mustafa Kemal C. / Kennedy C.	
18	Kennedy C. - Ahırkapı Feneri	Deri ve Tenasul H. H.
19	Divanyolu C. / Alemdar C.	Kuduz H.
20	Galata Köprüsü Eminönü Ayağı	
21	Atatürk Köprüsü Unkapani Ayağı	
22	Atatürk Bulvari / Şehzadebaşi C.	Özel Fatih H.
		Hifzisihha Enst.
23	Aksaray Meydanı	Doğaner H.
24	Millet C. / Adnan Adıvar C.	Haseki H.
25	Millet C. / Muska S.	Çapa Tıp Fakültesi H.
		Çapa Özben H.
26	Topkapı K. K.	
27	Vatan C. K. K.	

28	Ulubatlı Hasan K. K.	
29	Vatan C. / A. K. Bilek S.	Vakıf Gureba H.
		Topkapı H.
30	Vatan C. / Şehit Pilot Mahmut Nedim S.	Vatan H.
31	Cerrahpaşa C. / K. M. Paşa C.	Cerrahpaşa Tıp Fakültesi H.
32	Şehzadebaşı C. / Süleymaniye C.	Süleymaniye Doğumevi
		Esnaf H.
33	Topkapı - Edirnekapi C. / Fevzipaşa C.	
34	BKCY / Rami Kişla C.	
35	Fatih Köprüsü Ayvansaray Ayağı	
36	Ayvansaray C. / Demirhisar C.	Musevi H.
37	Sağmalcılar K. K.	Sağmalcılar Devlet H.
		Özel Bayrampaşa Merkez H.
38	Esenler K. K.	
39	Halkali Y. / TOY	
40	Mahmutbey K. K.	
41	Karadeniz K. K.	
42	Hasdal K. K.	
43	Silahtarağa C. / Abdi Efendi S.	SSK Eyüp H.
44	Sadabat Viyaduğu	
45	Okmeydani K. K.	
46	BKCY / Darulaceze C.	Darulaceze H.
47	Çağlayan K. K.	SSK Okmeydani H.
		Bulgar H.
48	Fatih Sultan Minberi C. / Paşakapısı C.	Kasımpaşa Deniz H.
49	Atatürk Köprüsü Tersane Ayağı	
50.	Karaköy C. / Kemeraltı C.	Beyoğlu H.
		St. George H.
51	Necatibey C. / Tophane İskele C.	Denizcilik Bankası H.
		İtalyan H.
52	Dolmabahçe C. / Kadırgalar C.	
53	İnönü C. / Şefikbey S.	Gümuşsuyu Askeri H.
54	Taksim Cumhuriyet Meydani	Taksim H.
		Alman H.
55	Tarlabaşı Bulvari / Ömer Hayyam C.	Tepebaşı Vatan H.
56	Cumhuriyet C. / Askerocağı C.	Fransız Pastor H.
		Ermeni Katolik Surp Agop H.
57	Cumhuriyet C. / Valikonağı C.	

58	Valikonağı C. / Rumeli C.	Teşvikiye Sağlık Evi SSK Kadın Doğum Kliniği A. Bristol H. Güzelbahçe Kliniği Pakize Tarzi Kliniği
59	Rumeli C. / Halaskargazi C.	Hayvan H.
60	Abide-i Hurriyet C. / Halaskargazi C.	Şişli Etfal H. SSK Şişli H. Ataman Kliniği Can H. Türk Kalp Vakfi Osmanoğlu Kliniği Fransız La Paix H. Hayat H.
61	BKCY - Büyükdere C.	Aksoy H. Mecidiyeköy Çevre H.
62	Barbaros Bulvarı / Beşiktaş C.	
63	Barbaros Bulvarı / Beşiktaş BKBY	Sait Çiftci Kamu Sağlıgı Merkezi
64	Barbaros Bulvarı K. K.	TKV Onkoloji Merkezi
65	BKCY / Beşiktaş BKBY	
66	Muallim Naci C. / İkmalpaşa S.	Ortaköy Şifa Yurdu
67	Büyükdere C. / Levent FSMKBY	Özel Levent H.
68	Harp Akademileri K. K.	
69	FSMKCY / Levent FSMKBY	
70	Rumelihisarı-Baltalimanı C. / Baltalimanı Çayır S.	Baltalimanı Kemik H. H.
71	Maslak Kavsağı	
72	Maslak Yolu / İstinye C.	
73	İstinye C. / Çapari S.	İstinye Devlet H.
74	Haci Osman Kavsağı	
75	Tarabya Kavşaği	
76	Hacı Osman Bayırı / Kefeliköy C.	
77	Mesarburnu C. / Sular C.	İsmail Akgun Devlet H.
78	Beylerbeyi K. K.	
79	Çengelköy C. / Güzeltepe C.	
80	Alemdağ C. / Nato Y. C.	
81	Altunizade K. K.	Marmara Üniversitesi H.
82	Koşuyolu C. / Tophanelioğlu C.	Polis H. Validebag Provantoryum - Sanatoryum

		Validebağ Öğretmen H.
83	İstanbul Ankara Devlet Y. / Koşuyolu C.	Koşuyolu Göğüs H. H.
84	Çamlıca K. K.	Çamlıca Göğüs H. Askeri H.
85	Uzunçayır K. K.	
86	Nuh Kuyusu C. / Dr. Eyüp Aksoy C.	Zeynep Kamil H.
87	İstanbul Ankara Devlet Y. / Tıbbiye C.	Haydarpaşa Numune H.
		Haydarpaşa G. K. D. Cer. Merkezi
		Gülhane Askeri Tıp Akademisi H.
88	Kadıköy Altıyol Meydanı	Kadıköy Sağlık Merkezi
89	Bağdat C. / Recep Peker C.	
90	BKCY / Fahrettin Kerim Gökay C.	
91	Fahrettin Kerim Gökay C. / Dr. Erkin C.	Göztepe SSK H.
92	Şemsettin Günaltay C. / Yelkenli Değirmen C.	
93	Bağdat C. / Ethem Efendi C.	Hayvan H.
94	Semşettin Günaltay C. / Sahrayı Cedid İçerenköy C.	SSK Geriatri H.
		SSK Psikiyatri H.
95	Bağdat C. / Ali Nihat Tarlan C.	
96	Bağdat C. / Atatürk C.	
97	Bağdat C. / Tugayyolu C.	
98	Cevizli K. K.	Kartal Devlet H.
99	Maltepe K. K.	SSK Süreyyapaşa Sanatoryumu
		SSK Provantoryum ve Rehabilitasyon
100	Bostancı K. K.	Organ Nakli H.
101	E-5 / Sahrayi Cedid İcerenköy C.	PTT H.
102	Kozyatağı K. K.	
103	AOY K. K.	
104	FSMKCY / Alemdağ C.	
105	Kavacık Kavşağı	
106	Kavacık Çiftliği C. / Kanlıca - A.Hisarı C.	
107	Paşabahçe Çubuklu C. / Paşabahçe İskele C.	Beykoz H.
		Paşabahçe H.
108	Fevzipaşa C. / İsakağa C.	Beykoz Çocuk Göğüs H. H.

APPENDIX 5. AN EXAMPLE OF HOW TO USE CAVR :

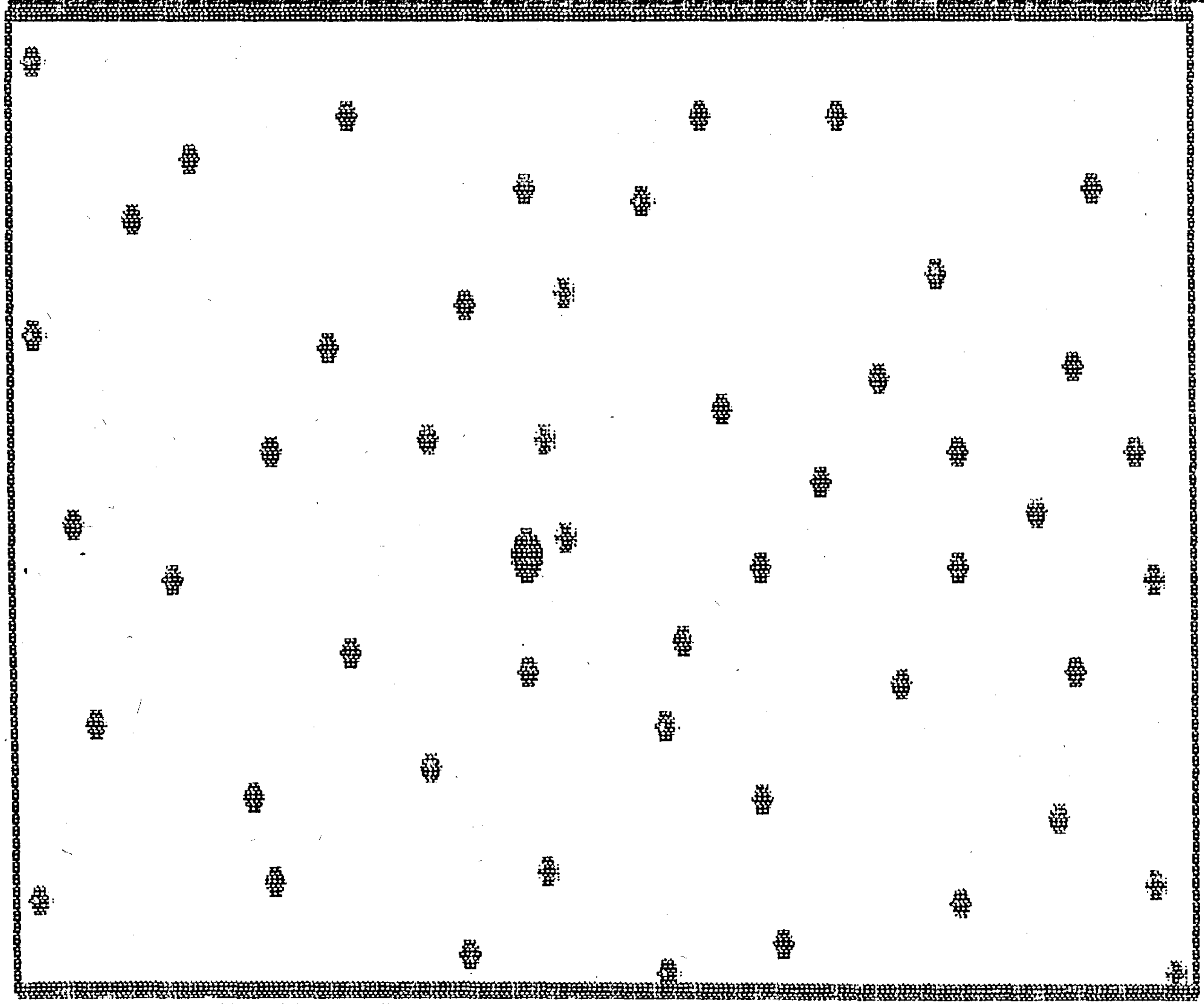
```
51              » Number of nodes plus one

X   Y    Q
30  40   0      » Coordinates and demand for node 1(depot )
37  52   7      » Coordinates and demand for node 2
49  49  30      »     "         "      "    "    "  3
52  64  16      »     "         "      "    "    "  4
20  26   9      .                                   .
40  30  21      .                                   .
21  47  15      .                                   .
17  63  19      .                                   .
31  62  23      .                                   .
52  33  11      .                                   .
51  21   5      .                                   .
42  41  19      .                                   .
31  32  29      .                                   .
 5  25  23      .                                   .
12  42  21      .                                   .
36  16  10      .                                   .
52  41  15      .                                   .
27  23   3      .                                   .
17  33  41      .                                   .
13  13   9      .                                   .
57  58  28      .                                   .
62  42   8      .                                   .
42  57   8      .                                   .
16  57  16      .                                   .
 8  52  10      .                                   .
 7  38  28      .                                   .
27  68   7      .                                   .
30  48  15      .                                   .
43  67  14      .                                   .
58  48   6      .                                   .
58  27  19      .                                   .
37  69  11      .                                   .
38  46  12      .                                   .
46  10  23      .                                   .
61  33  26      .                                   .
62  63  17      .                                   .
63  69   6      .                                   .
32  22   9      .                                   .
45  35  15      .                                   .
59  15  14      .                                   .
 5   6   7      .                                   .
10  17  27      .                                   .
21  10  13      .                                   .
 5  64  11      .                                   .
30  15  16      .                                   .
39  10  10      .                                   .
32  39   5      .                                   .
25  32  25      .                                   .
25  55  17      .                                   .
48  28  18    » Coordinates and demand for node 50
56  37  10    »     "         "      "    "    "  51
```

ALI TULMER AKYUZ ' 1991

OPTIONS

File Name:50_NODE

ALI TUNER AKYUZ ' 1991

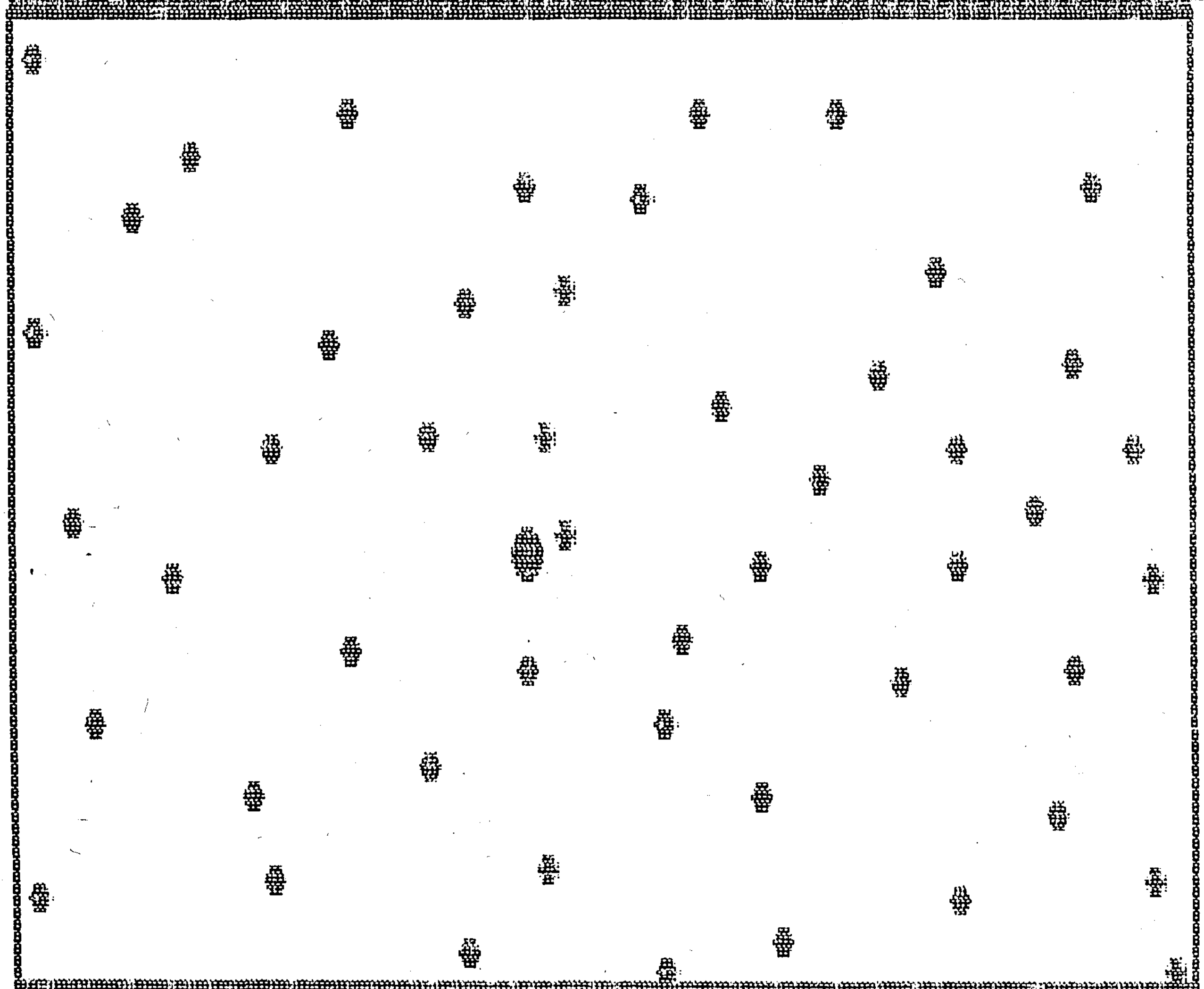


OPTIONS

Number of nodes : 50

create cost matrix (Y/N)?

ALI TUNER AKYUZ ' 1991

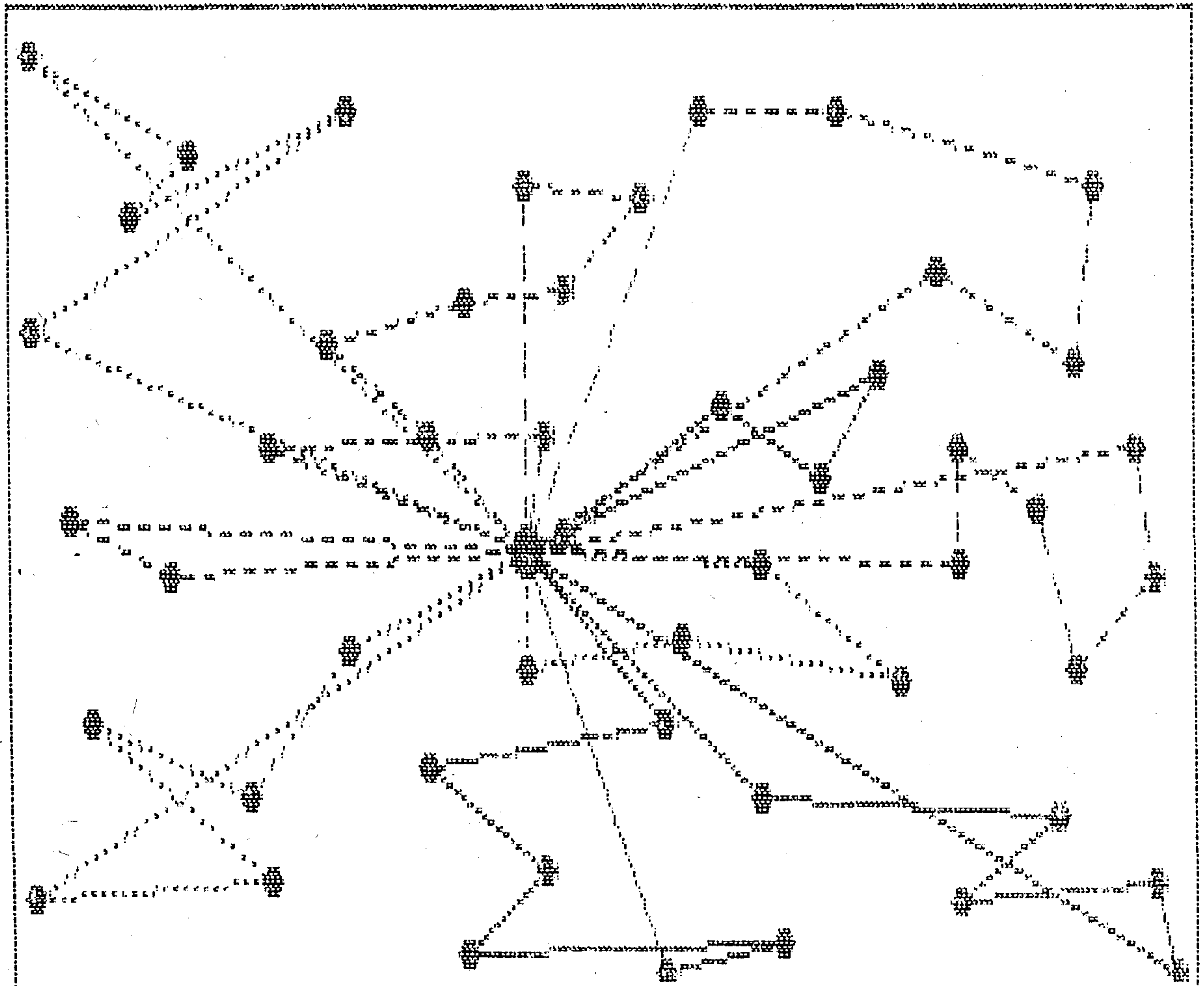


OPTIONS

Number of nodes : 50

finding initial solution...

ALI TUMER AKYUZ ' 1991



OPTIONS

Number of nodes : 50

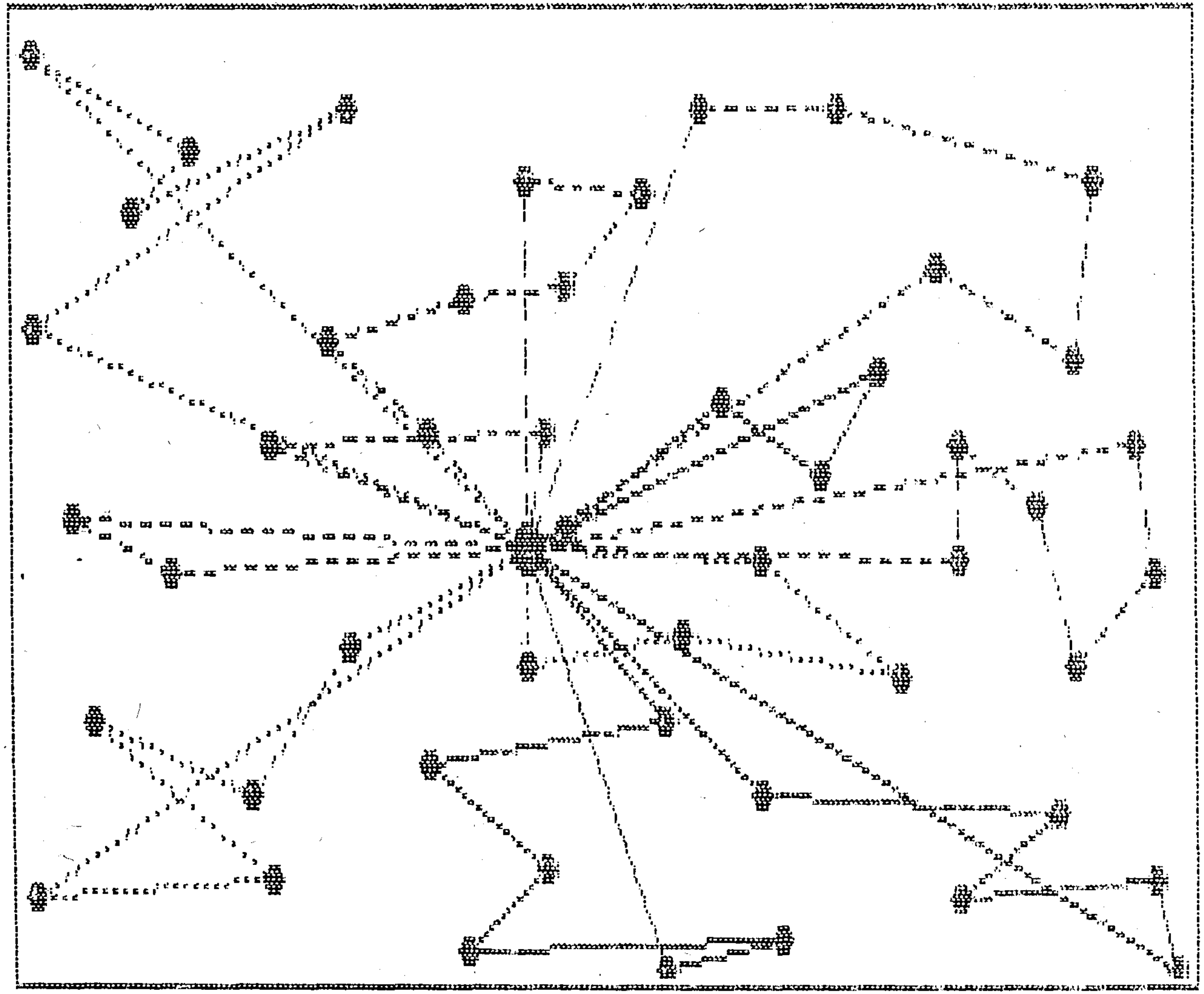
Total Cost : 861

Enter Pairs :
(0,0 to End...)

Pair[,1]=

Partial optimization (Y/N)?Y

ALI TUMER AKYUZ '1991



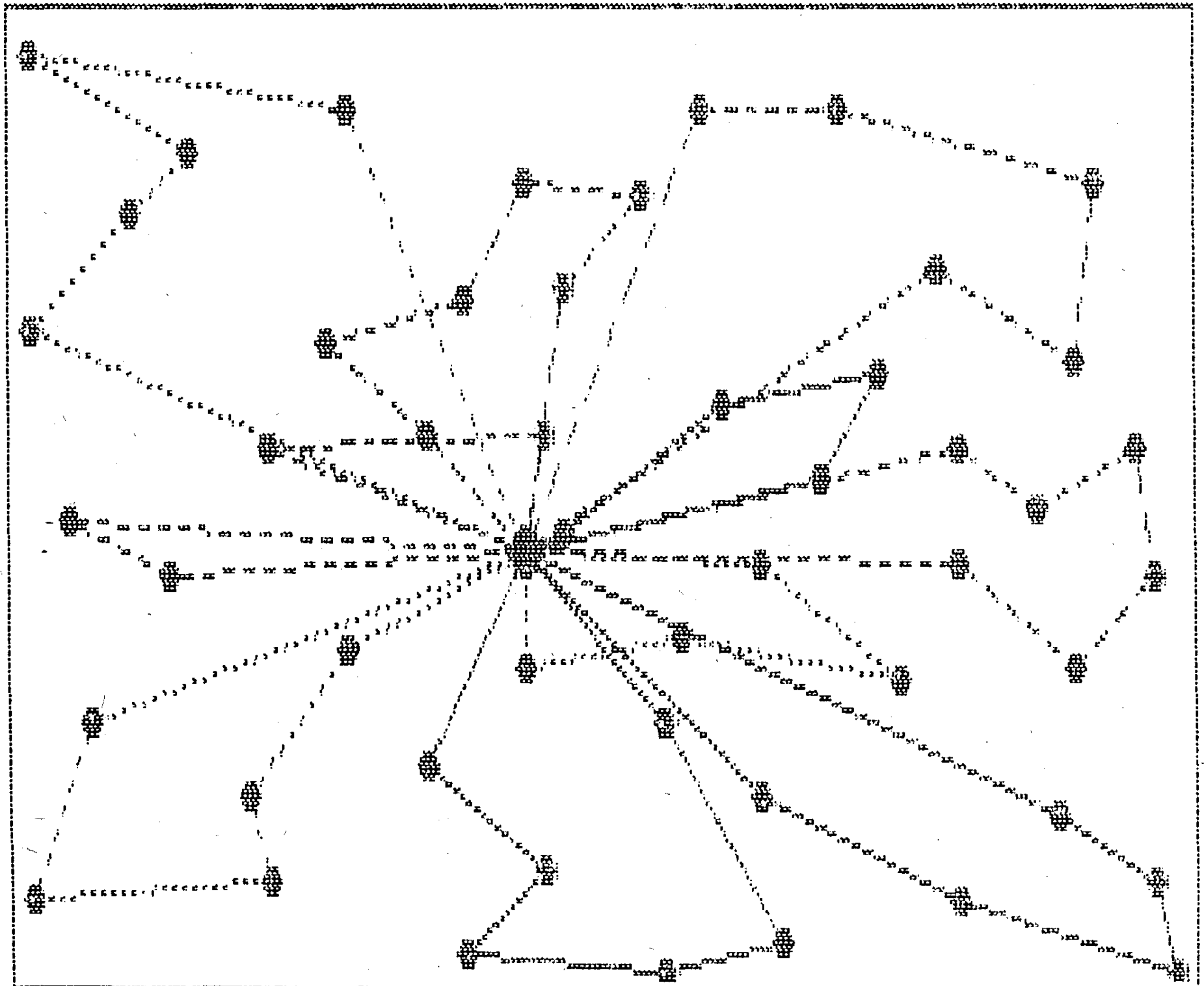
OPTIONS

Number of nodes : 50

Total Cost : 861

single route improvement...

ALI TUNER AKYUZ ' 1991



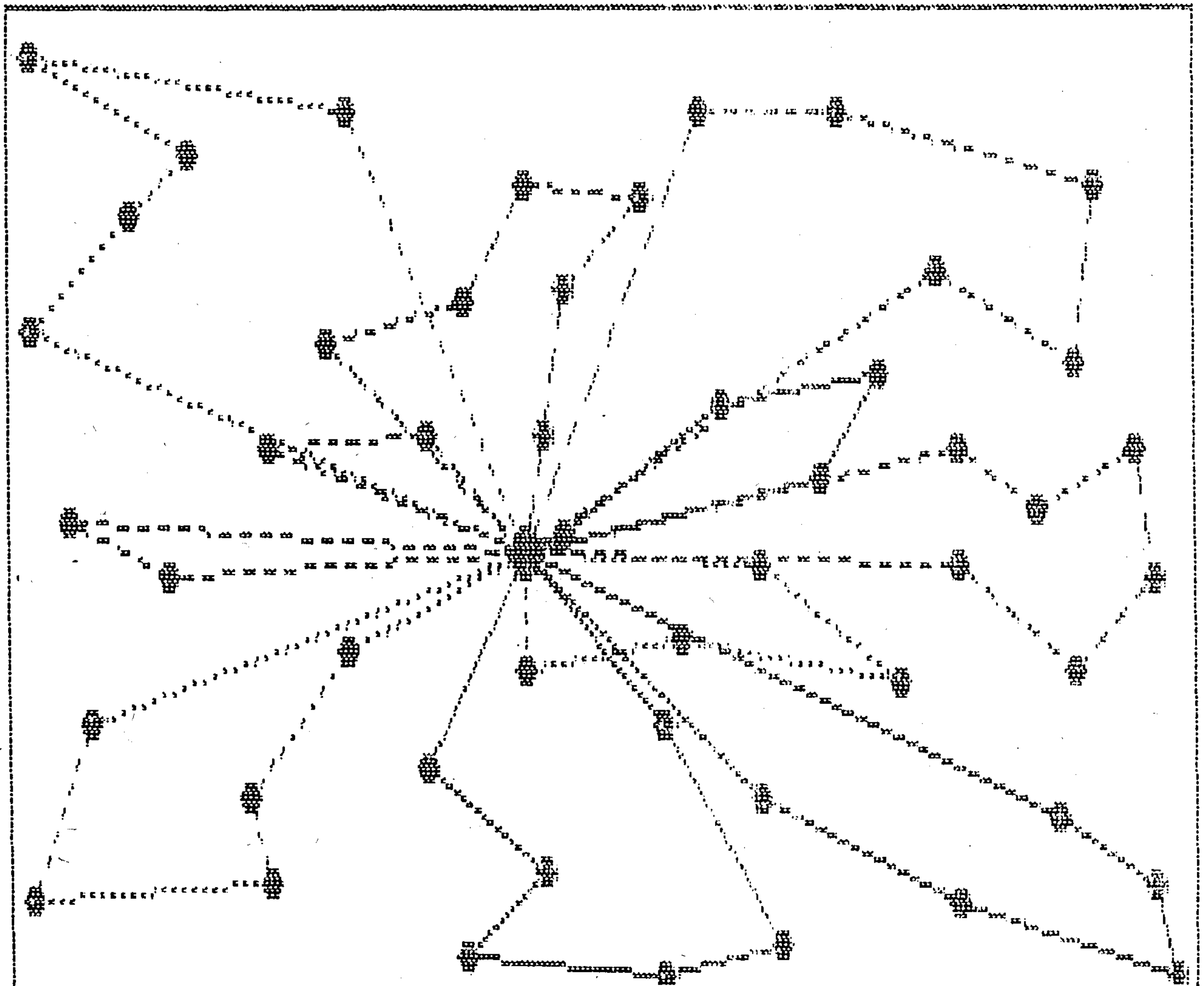
OPTIONS

Number of nodes : 50

Total Cost : 774

calculating one-node interchange savings...

ALI TUNER AKYUZ '1991



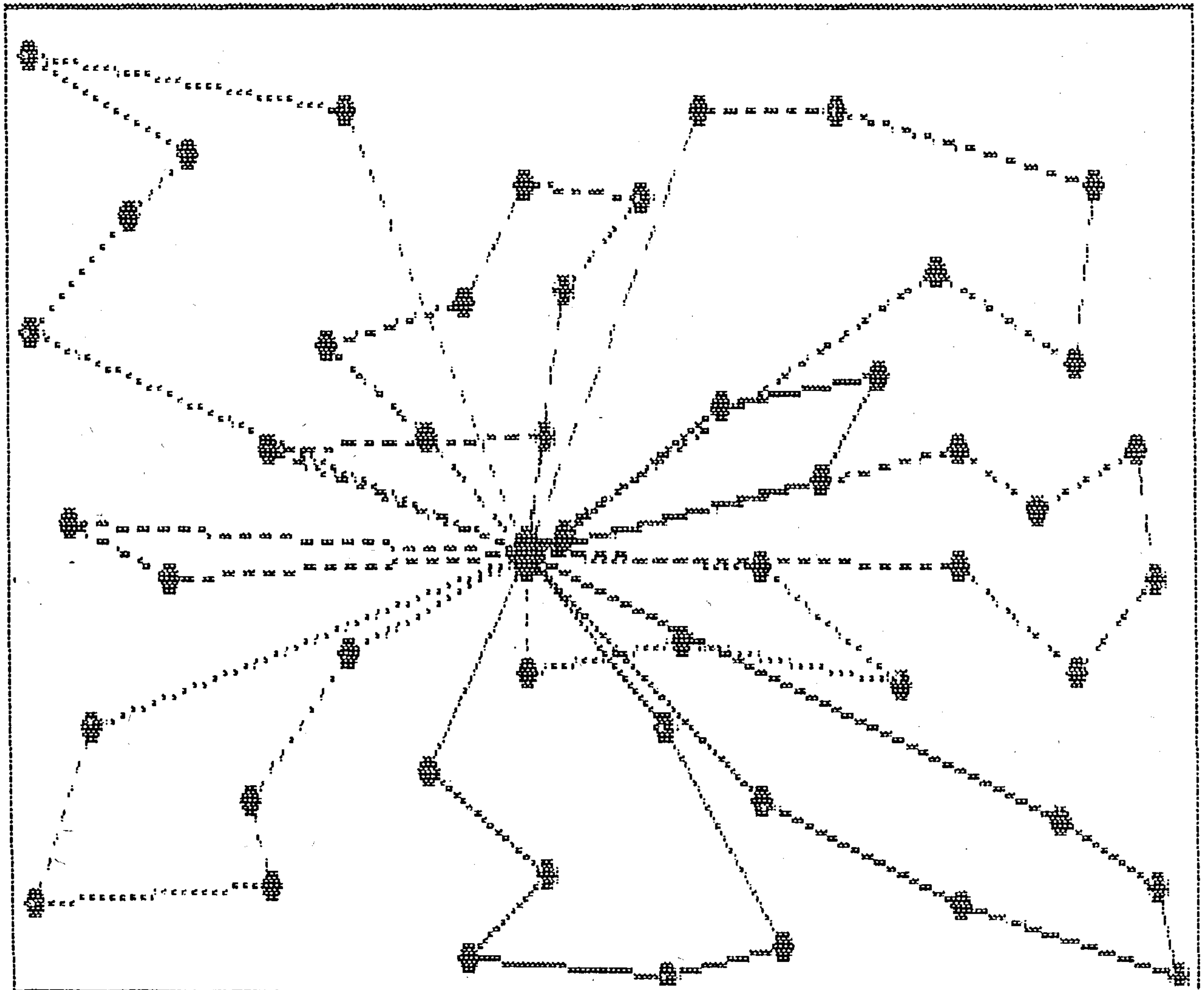
OPTIONS

Number of nodes : 50

Total Cost : 769

calculating two-node interchange savings...

ALI TUNER AKYUZ ' 1991

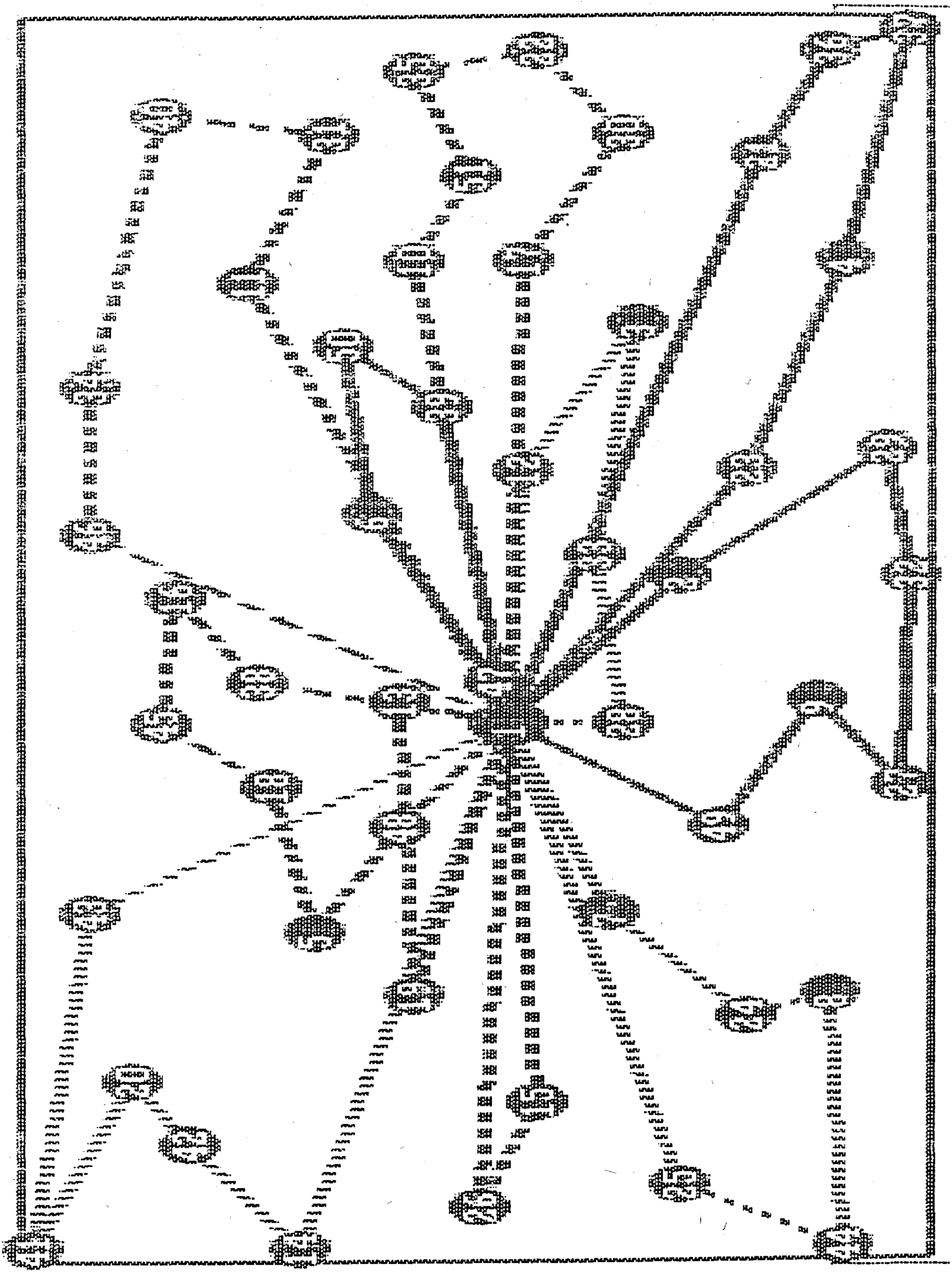


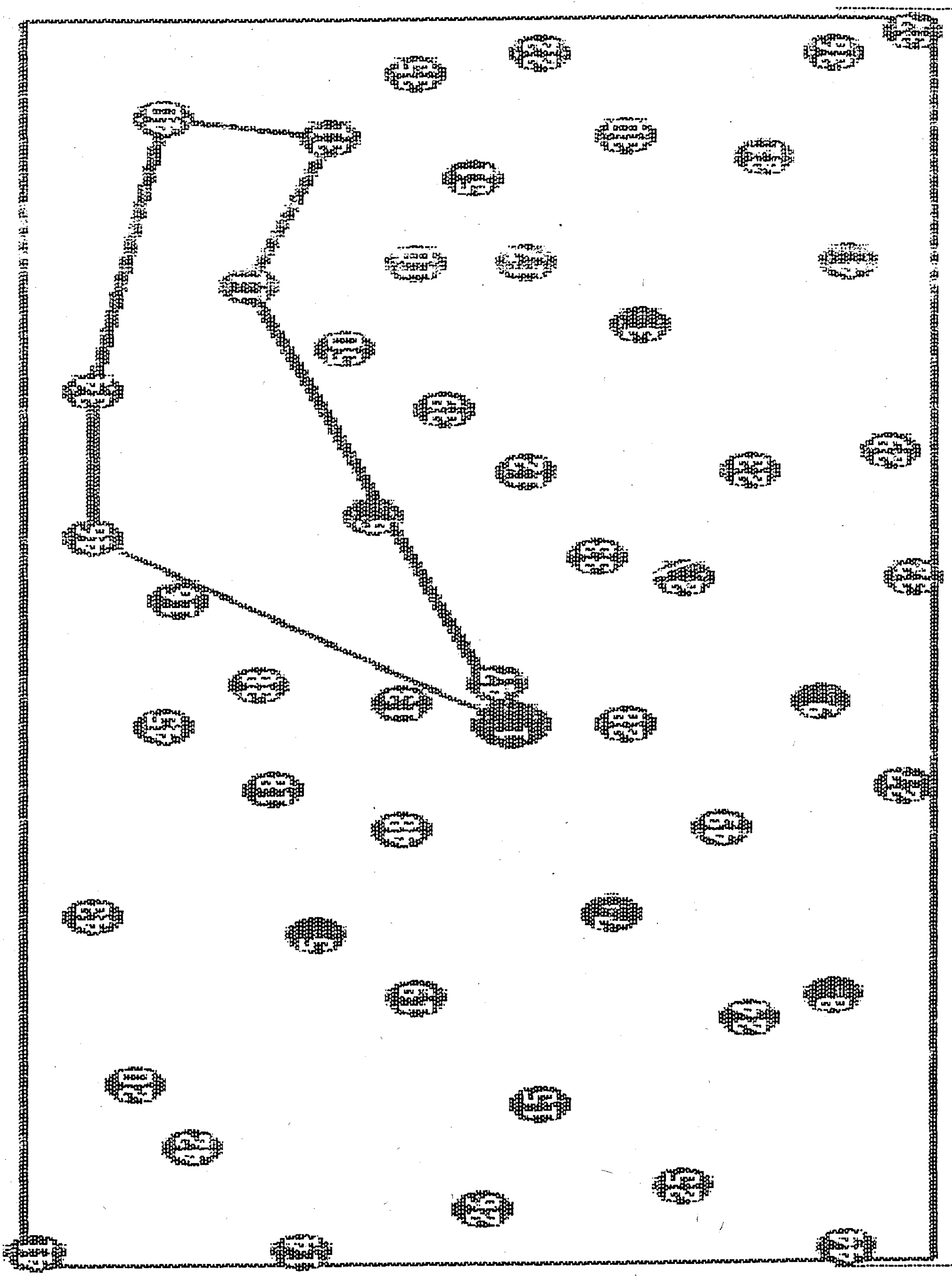
OPTIONS

Number of nodes : 50

Total Cost : 774

Z = Zoom   O = Zoom only one route   R = Report   I = Node Info   Enter =





ROUTES	VEHICLE CAPACITY	ROUTE COST	ROUTE DEMAND
21-36-37-4-23-1)	80	90	75
43-41-20-42-14-1)	80	101	79
46-34-40-31-11-1)	80	101	71
10-51-35-22-30-17-1)	80	82	76
49-9-27-32-29-2-1)	80	78	79
25-44-8-24-7-1)	80	77	71
38-16-45-18-5-48-1)	80	65	72
26-15-1)	80	47	49
39-50-6-47-1)	80	46	59
12-3-33-28-1)	80	50	76
19-13-1)	80	37	70

ber of Vehicles Used : 11
AL COST : 774

Press any key to return to graphics screen...