AN ABSTRACT OF THE THESIS OF

Daryl Stanley Lovro for the degree Master of Science
(Name) (Degree)
in Industrial Engineering presented on 4 Sept 1974
(Major) (Date)

Title: A BRANCH AND BOUND ALGORITHM APPLIED TO FIELD OFFICE LOCATION

Redacted for privacy

Abstract approved: ______________________
(Signature)
Dr. James L. Riggs

This thesis develops a model for locating field offices for the Motor Vehicles Division of Oregon. The locations are determined by minimizing the total cost to the Public. This is reasonable because the Public finances the operation and the opening of the offices through tax dollars, and it bears the expense of traveling to the offices to register vehicles and obtain licenses.

A branch and bound algorithm for warehouse location developed by Basheer Khumawala is applied to the field office location problem to determine the optimal locations. It was found that the algorithm runs quite efficiently, but the storage capacities needed to determine optimality are prohibitive for large problems. The storage problem was avoided by dividing the State into four areas and running each area separately. A modification in the computer code is suggested so that the algorithm works like a heuristic procedure. The solutions obtained are not guaranteed to be optimal, but much less storage is used to find the solution.

Several different costs for opening offices and for traveling were used to investigate the sensitivity of the locations.

The results from the study are encouraging and are presently being used by the Motor Vehicles Division to assist in determination of new office locations.

A Branch and Bound Algorithm Applied to Field Office Location

by

Daryl Stanley Lovro

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1975

APPROVED:

Redacted for privacy

Professor and Department Head of Industrial & General Engineering
in charge of major

Redacted for privacy

Dean of the Graduate School

Date thesis is presented 4 Sept 1974

Typed by Vicky Bliss for Daryl Stanley Lovro

ACKNOWLEDGEMENTS

I would like to thank Dr. James L. Riggs for his time, assistance, and helpful criticism.

Harvey Ward, Director of Field Services for the Motor Vehicles Division, must receive recognition for proposing this study and for aiding in the location of needed information. Also, Harvey Ward and the staff on the DMV Data Processing Center must be thanked for the computer time arrangements and for the assistance they provided.

Finally, my wife, Nancy, deserves thanks for her help and assistance in writing the thesis.

TABLE OF CONTENTS

LIST OF FIGURES

List of Figures (cont.)

LIST OF TABLES

A BRANCH AND BOUND ALGORITHM APPLIED TO FIELD OFFICE LOCATION

I. INTRODUCTION

How many field offices should the Motor Vehicles Division of Oregon provide to serve the public best? Where should these offices be placed? How large a staff should each office contain? In other words, a model is needed to optimize the services that the Motor Vehicles Division (DMV) can provide to the people of Oregon. These services involve the issuing of Drivers' Licenses and Vehicle Registrations. The Director of Field Services, Harvey Ward, provided information about the problem. He specified that the present locations should not be considered as constraints to finding the optimal locations. The purpose of this paper is to present a method, to derive a solution, and to investigate the feasibility of the results.

The DMV has about 45 field offices located throughout the State with the head office in Salem, Oregon. They handle Vehicle Registration, Driver Licensing, Public Utility Commission business, and Highway business. Only Vehicle Registration and Drivers License business will be considered in this paper. The business is handled partly through mail which is sent to the head office and partly through direct contact with the customer who comes to the field office. With the present field office locations, about 50% of the transactions occur at the field offices.1/ If there were fewer

1/ A transaction is considered the registration of a vehicle or the licensing of a driver.

field offices located in the State, it is possible that more of the transactions would be conducted through the mail -- people would not want to travel the extra distance. But, not all of the business can be accomplished through the mail, therefore, all the field offices cannot be eliminated. Even though it is possible for transaction levels to change with the relocation, addition, or elimination of field offices, it will be assumed that the levels remain at 50% through mail and 50% through the field offices.

A mathematical model will be used to solve the problem. Therefore, criteria which can be evaluated quantitatively need to be determined. Both tangible and intangible criteria should be included. ReVelle, Marks, and Liebman (1970) surveyed several methods for finding the location of facilities in both private and public business. They state that the criteria for evaluation costs in private business is more easily defined than in public business. In private business, locations can be determined by minimizing the total cost of operations. This approach compares the cost of opening a facility to the cost of travel resulting from going to another facility. With public operations, it is more difficult to evaluate the costs. If they cannot be determined, surrogates for utility are often used. For example, the objective of a model may be to minimize the total miles traveled to a facility given that there are a specific number of facilities.

In the problem discussed in this paper the desire is to find the optimal number of field offices and their locations. The question then arises, for whom are the locations being determined -- the Motor

Vehicles Division or the people of the State. The optimal policy for the DMV may be to open one office in the middle of the State and make each person who cannot do his business by mail travel to the office. This may be feasible, but is not practical. A more appropriate solution is to locate offices in positions where they are best for the majority of those concerned, mainly the drivers and car owners of Oregon. This is logical since it is the public's tax dollars which are used to operate the DMV, and it is the public's personal money that finances trips to the field offices. If reasonable costs can be determined, then a mathematical model can be set up to minimize the total cost to the public.

There are several factors which should be considered in determining field office locations. They are:

1. How far must the customers travel to the field offices?
2. How large is the demand for services?
3. What traveling expenses are incurred?
4. What is the cost of the public's time and inconvenience?
5. What are the operating expenses for the field offices?
6. What is the cost of opening an office?

These factors and their effects on the number and location of offices are shown quite clearly in Figure 1.1. As the number of offices increases, the travel cost decreases and the opening cost increases. The total cost is shown as the sum of the two cost functions. It can be seen that the objective of a location algorithm is to find the number and location of offices which minimize the total cost.

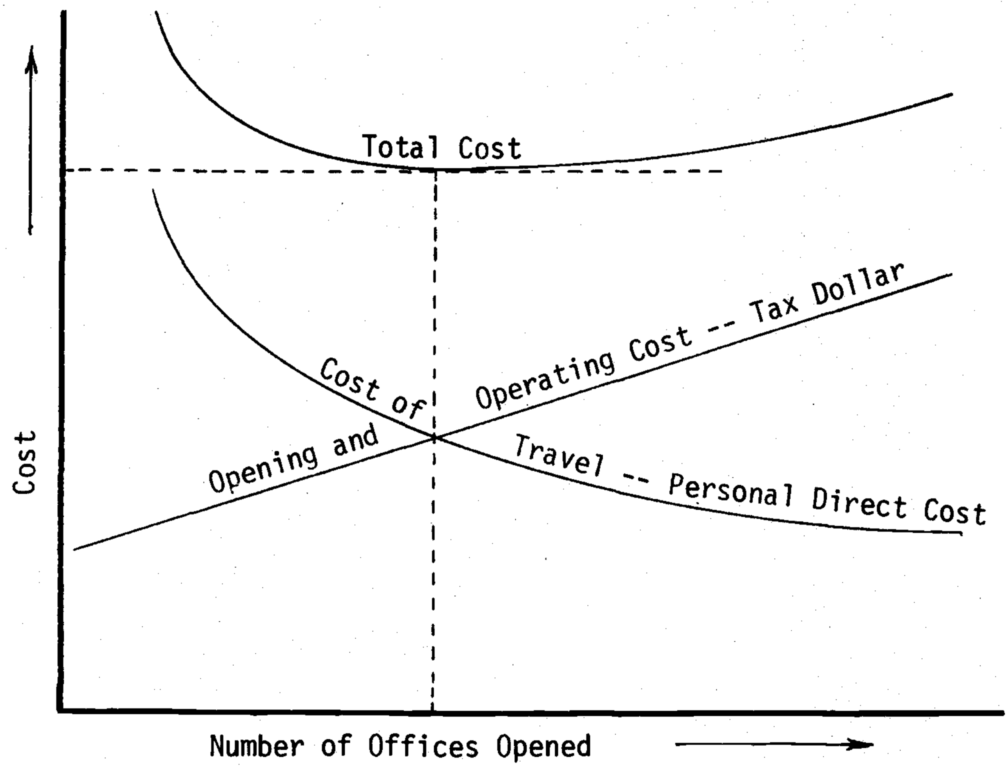


Figure 1.1. Relationship of field office location costs

The field office location problem comes from a group of problems associated with location analysis. Similar problems which use the same theory are the plant location problem, the fire and police station location problem, and the health facility location problem. The warehouse location problem or "simple" plant location problem as Spielburg (Jan.-Feb. 1969) puts it, is not a difficult one to formulate; but it does have combinatorial problems. The "simple" is added because of the assumption that each possible plant location is capable of supplying the total demand. The uncapacitated assumption is somewhat unrealistic in most cases, but it does lessen the difficulties of computation. The computational problem arises because a plant must either be opened or closed -- there can be no partially opened offices. Therefore, the problem comes into the category of mixed integer programming, zero-one programming, or fixed charge programming.

The following chapters will explain more fully what has been introduced here. Several algorithms will be evaluated, and then a description of the chosen procedure will be discussed. Also included, will be a discussion of the selection of data, and finally, the analysis of that data.

II. SEARCH FOR AN ALGORITHM

A literature search was made to find an algorithm which would run efficiently on a computer. Efficiency is important because of the size of the field office problem -- it originally has 417 cities and 114 possible office locations. All of the methods investigated have formulations which could be adapted to the problem. Some fit better than others. The algorithms in Table I were investigated. Each of the researchers added their own individualities to the algorithm.

TABLE I. PAGE NUMBERS OF ALGORITHMS

		ALGORITHMS INVESTIGATED			
		Direct Search	Linear Programming	Heuristics	Branch & Bound
RESEARCHERS	Abernathy & Hershey	p. 7			
	Keuhn & Hamburger			p. 10	
	Feldman, Lehrer, & Ray			p. 11	
	Revell & Swain		p. 8		
	Efroymson & Ray				p. 14
	Kurt Spielburg		p. 9		p. 15
	Basheer Khumawala				p. 16

Direct Search

Using a direct search involves investigating many of the possible solutions to a problem and then picking the best one. For example, if one has a map with several mountains and he wants to find the two highest peaks using a computer, a grid would be superimposed on the map. The routine would probably start at one corner investigating the altitude at each point on the grid. It would continue the investigation until it found the two highest peaks. It is a very time consuming procedure. Heuristics can be used to minimize the number of points investigated.

Abernathy and Hershey (1972) did an interesting study on planning the location of Regional Health services. Their formulation took into account three factors: (1) utilization of the health center, (2) the distance to the center per person, and (3) the distance to the center per encounter. These location criteria provide a means of evaluating the needs of the people and were of more interest to the authors than the method used to solve the problem. They used a direct search algorithm developed by Hooke and Jeeves (1961). This procedure makes use of a large amount of computer time and storage to find an optimal solution. Thus, it limits the size of the problem which can be handled.

Linear Programming

Linear Programming (LP) is very popular for optimizing convex functions. It reaches a solution rapidly compared to other methods of optimization, but it assumes linearity and continuity.

Revelle and Swain (1970) worked the problem locating a given number of m facilities in n communities. The objective of their formulation is to minimize the number of miles that the total population travels. The formulation structured as an LP problem is:

$$\text{minimize:} \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot d_{ij} \cdot x_{ij}$$

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

$$x_{jj} \geq x_{ij} \qquad i = 1,2,\ldots,n; \quad j = 1,2,\ldots,n; \quad i \neq j$$

$$\sum_{i=1}^{n} x_{ii} = m \qquad x_{ij} \geq 0 \qquad i = 1,2,\ldots,n; \quad j = 1,2,\ldots,n$$

where:

a_i = population

d_{ij} = the distance between i and j

m = the number of facilities

n = the number of communities

x_{ij} = the fraction of a community, i, assigned to facility j. In the optimal solution, x_{ij} is equal to 0 or 1.

This formulation is appropriate for field office location. It requires the number of desired office locations be given. The present number of offices operated by the DMV could be used as the m value. The deficiency is that the optimum m value is not identified. Therefore, a cost analysis would have to be made to determine the optimal number of offices to open. This would be very difficult to evaluate, because it cannot be assumed that the optimal number of field offices for the DMV is the optimal number offices for the people of Oregon. Somehow one must evaluate the needs of the people and the needs of the DMV together.

Linear programming does not guarantee integer solutions. Thus, the value x_{ij} is not always a 0 or 1 integer. The authors say that it is unusual for a fractional result to occur. If it does occur, however, a branch and bound technique is recommended to find the optimal solution.[2/]

Since the solutions resulting from the LP are optimal whether the results are fractional or integer, the results could be used to check the solutions obtained from a heuristic program. Should the heuristic solution be near the optimum, then the facility assignments can be assumed to be reasonable.

Kurt Spielburg (Jan.-Feb. 1969) has done most of his work with branch and bound algorithms, but suggests that the formulation shown by equations 2.5 to 2.9 can be solved by using linear programming. This can be done by weakening the constraint $Y_i = 0$ or 1 to $Y_i \geq 0$ and $Y_i \leq 1$. This method, similar to the Revell and Swain (1970)

2/ The solution is optimal given that the value for m is optimal.

approach, does not guarantee an integer solution. Spielburg's formulation provides a better means of finding a solution for the field office problem than does Revell and Swain's formulation. Spielburg's formulation minimizes the total cost of an operation which in this case includes the operations of the DMV and the travel expenses incurred by the people of Oregon. Minimizing the total cost obtains a more representative solution for all those concerned.

Heuristics

Heuristics are a set of rules or guidelines which are used to find a solution to a problem. Using heuristics can avoid some of the problems found in optimizing procedures. Two of the main problems are the amount of storage capacity needed and the length of the computing time. A heuristic procedure works toward a solution which is acceptable in terms of the characteristics of the program, but is not necessarily optimal.

Kuehn and Hamburger (1963) were pioneers in the use of a heuristic approach for solving the location problem. Their program has two parts: "(1) the main program, which locates warehouses one at a time until no additional warehouses can be added without increasing the total cost, and (2) the bump and shift routine, entered after processing in the main program by evaluating the profit implications of dropping individual warehouses or of shifting them from one location to another." (Kuehn, 1963. p. 645) They used three heuristics:

1. The warehouse will be in locations where the demand has the greatest concentration. Therefore, many geographical locations can be eliminated from consideration.

2. Near optimum solutions can be arrived at by adding warehouses which produce the greatest cost saving, one at a time.

3. Only a small portion of the possible warehouse locations need to be evaluated when determining the next location.

Kuehn and Hamburger's computational experience is based on a problem with 50 customer locations and 24 potential warehouse locations. Twelve possible cases were evaluated. The program produced near optimum results in an average running time of two minutes, 30 seconds on the IBM-650.[3/] Running time appears to increase linearly with the number of warehouses times the number of customers.

Kuehn and Hamburger suggested a program be set up which would eliminate warehouses one by one based on cost savings rather than adding the warehouses one by one. This procedure would be more efficient in some cases; for example, when the number of warehouses located is more than half the number of potential warehouses. Feldman, Lehrer and Ray (1966) look at this approach.

Feldman, Lehrer and Ray (1966) follow the Kuehn and Hamburger approach. There are two basic differences in the methods:

1. Feldman, Lehrer and Ray extended their heuristics to handle concave F_i, the cost of opening a warehouse.

2. They "drop" warehouses instead of add them.

3/ Running time on different computers is hard to compare since there are so many types, combinations, and improvements. Therefore, times should not be taken too seriously.

They evaluate F_i as a concave function which varies with the size of the warehouse. It is cheaper per unit to open a large warehouse than it is a small one. This is interesting because most formulations consider F_i as a constant opening cost.

Feldman, Lehrer and Ray suggest that the "drop" routine is better than Kuehn and Hamburgers because it is more convenient when forbidden shipping routes occur. Also, companies are rarely interested in building from scratch, rather they want to eliminate.

The computer code was tested using problems which Kuehn and Hamburger solved. The authors then found their own solutions were as good as Kuehn and Hamburger's. The CPU time on an IBM 7094 was under one minute. Following this, a much larger problem was investigated. It was found that the solution obtained by their drop routine had a cost which was only 0.5% greater than the optimal. Thus, the heuristic provided warehouse locations which were acceptable.

Branch and Bound

> "The branch and bound methods are enumerative schemes for solving optimization problems. The utility of the method drives from the fact that, in general, only a small fraction of the possible solutions needed actually be enumerated, the remaining solutions being eliminated from consideration through the application of bounds that establish that such solutions cannot be optimal." (Mitten, 1970. p. 24)

The procedure is implied by the name -- first you branch then you bound. Before this procedure starts, the linear programming problem is solved to see if the solution meets the integer constraints. Suppose that the constraints require Y_i to be equal to 0 or 1.

If so, the solution is optimal and the algorithm terminates. If not, branching begins, and a branch and bound tree (Figure 2.1) is constructed. Two branches emanate from the first node. On the first branch, one of the noninteger variables Y_i' is forced to zero. The resulting solution is Z_1. On the second branch Y_i' is forced to one. Its solution is Z_2. These solutions must either be terminal solutions, solutions which meet the integer constraints, or nonterminal solutions, solutions which do not meet the integer constraints. Now, the bounding begins. $Z' = \text{Min}\ (Z_1, Z_2)$. If Z' is nonterminal then it is compared with the current lower bound (LB). If $Z' < \text{LB}$ then $\text{LB} = Z'$.

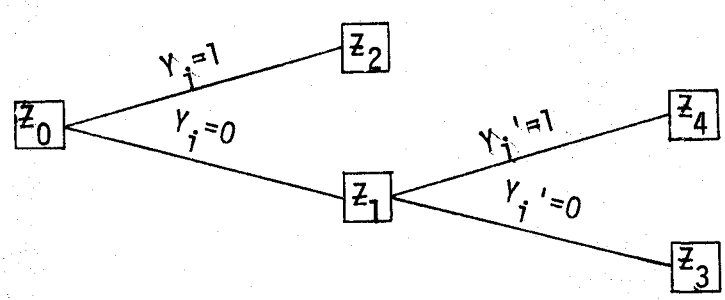


Figure 2.1. Branch and bound tree.

Branching begins again by branching from a nonterminal node (solution) with a solution less than the current upper bound (UB). The branches result in solutions Z_3 & Z_4. If $Z'' = \text{Min}\ (Z_3, Z_4)$ is a terminal solution, then Z'' is compared with UB. If $Z'' < \text{UB}$, then UB = Z''. When a terminal solution is reached, no further branches can emanate from it. No branching can occur at a nonfeasible[4/] node either. The process of branching and improving the bounds ends when all nodes with solutions less than the current upper bound have been investigated. The optimal solution is then the current upper bound. Another interpretation is that the optimal solution is the minimum of all the terminal nodes.

There are several problems with branch and bound procedures. The computer time is usually quite high because of the number of LP problems that must be solved. This also causes a storage problem because of the number of solutions that must be kept in order to compare the results.

Efroymson and Ray (1966) reformulated the model shown in equations 2.5 to 2.9 because the linear programming problem must be solved so many times in a branch and bound algorithm. As a result, the LP problem can be solved more efficiently.

In this formulation, N_j is the set of offices which can supply customer j, and P_i is the set of customers who can be supplied from plant i. The reformulation is:

$$\text{minimize:} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} + \sum_{i=1}^{n} F_i Y_i \qquad (2.1)$$

$$\text{subject to:} \quad \Sigma_{i \varepsilon N_j} \; X_{ij} = 1 \qquad j = 1,2,\ldots,n \quad (2.2)$$

$$\Sigma_{j \varepsilon P_i} \; X_{ij} \leq n_i \; Y_i \qquad i = 1,2,\ldots,m \quad (2.3)$$

4/ A nonfeasible node is a node which has at least one demand center that cannot be serviced by an open field office because of a prohibited route.

$$Y_i = 0 \text{ or } 1 \quad X_{ij} \geq 0 \qquad (2.4)$$

Where: C_{ij} = the cost for a demand center j to go to a facility i.

F_i = The opening cost.

Khumawala offers this same formulation, and it is discussed on page 21.

In reference to Efroymson and Ray's computational experience, they found that computer storage and computer time cause the most difficult problems. Therefore, they implemented the following features to minimize the storage and computer time:

1. If a good solution is known to the problem, then no nodes whose solutions are greater will be stored.
2. If a terminal solution (all Y_i's are 0 or 1) whose solution is less than all previous terminal nodes is found, the program terminates.

They worked problems with 50 warehouses and 200 customers with an average computer time on an IBM 7094 of about ten minutes.

Kurt Spielburg (Nov. 1969) has worked extensively with branch and bound algorithms for plant (warehouse) location. He found that one of the characteristics of branch and bound algorithms used in location problems is that they are efficient when the solution is close to the origin[5/] and inefficient otherwise. Thus, some problems can easily be solved if a solution is arrived at by starting with all the plants open, but are almost impossible to solve if the solution is arrived at by starting with closed plants. To try to avoid the problem, Spielburg

5/ At the origin all plants are initially all open or all closed. The procedure then closes or opens plants respectively.

developed an algorithm which permits the start of a search at any convenient point. It could start with a good solution which would be generated after a certain amount of preliminary computation.

Spielburg handled several different realistic problems. His results are encouraging. By using his generalized search method as opposed to the natural search method, the solution times are decreased significantly.

Basheer Khumawala (1972) improved the algorithm developed by Efroymson and Ray. To overcome problems of storage and computational time, Khumawala derived an improved method of solving the linear program and developed test branching decision rules for determining which free warehouse (a warehouse neither opened or closed) to branch on in the next iteration. He uses Efroymson's and Ray's simplification procedures to reduce the size of the branch and bound tree.

Khumawala's computational experience is not as extensive as Spielburg's but the results are valuable. Sixteen test problems of size (25 X 50) were used to test the effectiveness of the algorithm and the branching decision rules. It was found that the largest omega rule[6/] was best. The computation time averaged 3.8 seconds on a CDC 6500 for the largest omega rule. It was also noted that the efficiencies increase with a sparse C_{ij} matrix; that is, a matrix which has many prohibited routes.

6/ The largest omega rule says to open the facility from among the group of free facilities which has the largest omega Ω. The omega value is explained on page 23.

Comments on the Solution Procedures

The direct search procedure use by Abernathy and Hersey (1972) was eliminated almost immediately. It cannot handle a problem of the size being considered in this paper.

Linear programming could be used to find a solution, but does not guarantee integers. Thus, only parts of offices might be opened. One would have to resort to another method of solution to find the results. Since this is the case, it would probably be better to use another method such as heuristics or branch and bound.

The use of heuristics seems to be a reasonable approach for solving the office location problem. The main drawback is that the solutions are not necessarily optimal. Branch and bound procedures guarantee optimal solutions. The running times for the branch and bound procedures may be somewhat longer but with a high speed computer, there should be no problem. One of the branch and bound procedures will be used because it gives an optimal solution. If storage becomes a problem with a field office location, then it can be broken down into parts and solved separately.

The decision about whose branch and bound algorithm to use, Khumawala's (1972) or Spielburg's (Nov. 1969.), was a toss up. Spielburg's algorithm has a feature which Khumawala's does not have. It has the ability to make use of a previous solution or a good solution which is not optimal. This feature makes it possible to find an optimal solution to large problems which must have many nodes (possible solutions) investigated to find the optimum. Khumawala's algorithm appears to be more efficient, but it is hard to evaluate the difference unless the two algorithms are tested on the same problems. The final decision is to use Khumawala's algorithm because of the availability of his computer code[7/] and amount of time which it would take to write and debug a program using Spielburg's algorithm.

The formulation which Khumawala uses is very applicable to the field office problem. He minimizes the total cost like Spielburg. It is a more useful approach than minimizing the total miles traveled. In the end, the miles traveled are minimized with respect to the cost of opening a field office. The development of the formulation follows.

The Formulation

Many of the formulations for facility location problems are very similar to the one presented here. The initial model is one offered by Spielburg (Jan. - Feb., 1969. pp. 86-88). It is developed into the final model used for solving the field office problems.

There are $\underline{n}$ demand centers with a demand $D_j (j=1,2,...n)$, and $\underline{m}$ possible field office locations. A field office may or may not be opened. If it is opened, there is an opening cost or a fixed cost, $F_i \geq 0$, associated with it. If it is not opened, then the

---

7/ The computer code is shown in Appendix E.

cost is zero. In mathematical terms, $Y_i = 1$ if it is opened, and $Y_i = 0$ if it is closed. The value ξ_{ij} in the formulation below is the amount of service supplied by office i to meet the demands of center j. Each office is capable of meeting the demands of all the demand centers. The cost[8/] of meeting this demand is γ_{ij} which is the cost per unit. The objective of the formulation is to minimize the total costs of operations. It is:

$$\text{Minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}\, \xi_{ij} + \sum_{i=1}^{m} F_i\, Y_i$$

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

$$\sum_{j=1}^{n} \xi_{ij} \leq Y_i \cdot u_i \qquad i = 1,2,\ldots,m$$

$$Y_i = 0 \text{ or } 1 \qquad \xi_{ij} \geq 0$$

The u_i represents an upper bound which could be set equal to $\sum_{j=1}^{n} D_j$ independent of i. It permits office i to service demand center j if $Y_i = 1$ and does not permit it if $Y_i = 0$.

The first part of the objective function $\sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij}\, \xi_{ij}$ can be solved if the minimum transportation cost from demand center j to field office i is chosen. For this reason, the problem is reformulated into a simpler form. The ξ_{ij} are replaced by X_{ij} where $X_{ij} = \xi_{ij}/D_j$. The

8/ The cost includes transportation costs and operating cost.

X_{ij}'s are interpreted to be the fraction of the demand serviced by office i. Also, since the purpose of inequalities is to prevent a demand center j from being assigned to a closed office or permit it otherwise, it can be replaced by $\sum_{j=1}^{n} X_{ij} \leq Y_i \cdot n_i$. The value n_i is the number of demand centers which can be serviced by office i. The resulting formulation becomes:

$$\text{Minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} + \sum_{i=1}^{m} F_i Y_i \qquad (2.5)$$

Subject to:

$$\sum_{i=1}^{m} X_{ij} = 1 \qquad j = 1,2,...,n \qquad (2.6)$$

$$\sum_{j=1}^{n} X_{ij} \leq n_i Y_i \qquad i = 1,2,...,n \qquad (2.7)$$

$$Y_i = 0 \text{ or } 1 \qquad X_{ij} \geq 0 \qquad (2.8)$$

$$\text{where: } C_{ij} = \gamma_{ij} D_j \qquad (2.9)$$

The branch and bound algorithm requires that a linear programming problem be solved at each node. If many nodes must be investigated to determine the optimal solution, much computer time will be used solving the LP. The number of LP problems solved varies a great deal. It can be as few as one or as many as several hundred. Thus, the formulation is again modified to simplify the solution of the LP. The Efroymson and Ray (1966) formulation is repeated here for convenience.

$$\text{Minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij} + \sum_{i=1}^{m} F_i Y_i \qquad (2.1)$$

Subject to:

$$\Sigma_{i\varepsilon N_j} X_{ij} = 1 \qquad j = 1,2,\ldots,n \qquad (2.2)$$

$$\Sigma_{j\varepsilon P_i} X_{ij} \leq n_i Y_i \qquad i = 1,2,\ldots,m \qquad (2.3)$$

$$Y_i = 0 \text{ or } 1 \qquad X_{ij} \geq 0 \qquad (2.4)$$

N_j = the set of field offices which can supply demand center j.

P_i = the set of centers that can be serviced by office i.

Also for each node (solution) the sets K_0, K_1 and K_2 are defined.

K_0 = is the set of closed offices. Y_i's are the set equal to zero.

K_1 = is the set of opened offices. Y_i's are set equal to one.

K_2 = is the set of offices which are neither opened nor closed. They are free offices. Y_i's are fractional.

Discussion of the Branch and Bound Algorithm

The formulation of the location problem is quite simple. The main problem is computational since it comes into the area of integer programming.

The formulation is set up so that the LP problems can easily be solved for uncapacitated problems. Other than this modification, there are three simplification procedures which are presented by Khumawala (1972). They reduce the number of nodes that must be investigated. In other words, they reduce the size of the branch and bound tree.

1. The first simplification determines the minimum bound for opening a field office. If it is positive, then the office is fixed opened. In mathematical terms, this is:

$$\nabla_{ij} = \mathrm{Min}_{k \varepsilon N_j \cap (K_1 \cup K_2);\ \ k \neq i} \ [\mathrm{Max}\ (C_{kj} - C_{ij},\ 0)]$$

$$\Delta_i = \Sigma_{j \varepsilon P_i}\ \nabla_{ij} - F_i$$

"If $\Delta_i > 0$, then $Y_i = 1$ for all branches emanating from that node." (Khumawala, 1972. p. B-720) Delta (∇_{ij}) is the savings that results if office i is opened to service city j. If the sum of the deltas for for office i is greater than F_i, the cost of opening the office, then it pays to open the office.

2. The second simplification is mainly an updating procedure. It reduces n_i, the number of cities which are serviced by office i.

"If for $i \varepsilon K_2$, $j \varepsilon P_i$

$$\mathrm{Min}_{k \varepsilon K_1 \cap N_j} (C_{kj} - C_{ij}) < 0$$

then n_i is reduced by one." (Khumawala, 1972. p. B-721)

All this says is that if it is cheaper for demand center i to be serviced by an open field office than it is to be serviced by one of the free field offices at the node, then demand center i should not be considered as a possible customer of the free field offices.

3. The third simplification is similar to the first. Instead of determining if the cost savings warrants the opening of a field office, it determines whether the cost reduction resulting from an office being open is still warranted. Also, it determines whether a free office can be closed.

"For $i \varepsilon K_2$, $j \varepsilon P_i$

$$\omega_{ij} = \mathrm{Min}_{k \varepsilon N_j \cap K_1} [\mathrm{Max}\ (C_{kj} - C_{ij}, 0)]$$

$$\Omega_i = \Sigma_{j \varepsilon P_i} \ \omega_{ij} - F_i$$

"If $\Omega_i < 0$, then $Y_i = 0$ for all branches emanating from the node." (Khumawala, 1972. p. B-721) ω_{ij} is the minimum savings which result from having office i open and city j being serviced by it. If the sum of these savings for the office i is less than the cost of originally opening the office, F_i, the office is closed. These simplifications are cycled through each iteration.

The simplification procedure is shown step by step in Figure (2.3). The branch and bound procedure is shown in Figure (2.2) The flow chart for the main program will be used in the following explanation.

When no further simplifications can be made, then LP is solved. Khumawala (1970. pp. 46-49) presents a time saving method to solve the LP (step M-3). It simply selects the feasible offices which will minimize the objective function at the node. The solution is defined by the following sets:

S_1 = the set of demand centers best serviced by open offices.

$$=\left\{ j(i_1) \;\middle|\; \nabla_{i_1} j(i_1) \geq 0 \; ; \; i_i \varepsilon K_1 \cap N_{j(i_1)} \right\}$$

S_2 = the set demand centers best serviced by free offices.

$$=\left\{ j(i_2) \;\middle|\; \nabla_{i_2} j(i_2) \geq F_{i_2} / n_{i_2} \; ; \; i_2 \, \varepsilon \, K_2 \cap N_{j(i_2)} \right\}$$

$S_1 U S_2$ = the set of remaining demand centers.

The solution is determined by the following equations:

$$j(i_1) \; \varepsilon \; S_1 \quad \begin{cases} X_{i_1 j(i_1)} = 1 \\ X_{i \; j(i_1)} = 0 \quad i \neq i_1 \end{cases}$$

$$j(i_2) \; \varepsilon \; S_2 \quad \begin{cases} X_{i_2 j(i_2)} = 1 \\ X_{i \; j(i_2)} = 0 \quad i \neq i_2 \end{cases}$$

$$j \varepsilon S_1 U S_2 \quad \begin{cases} X_{ij} = 1 \text{ if } C_{ij} + F_{i/n_i} = \mathrm{Min}_{k \varepsilon K_1 U K_2} \left[C_{kj} + g_{k/n_k} \right] \\ X_{ij} = 0 \text{ otherwise.} \end{cases}$$

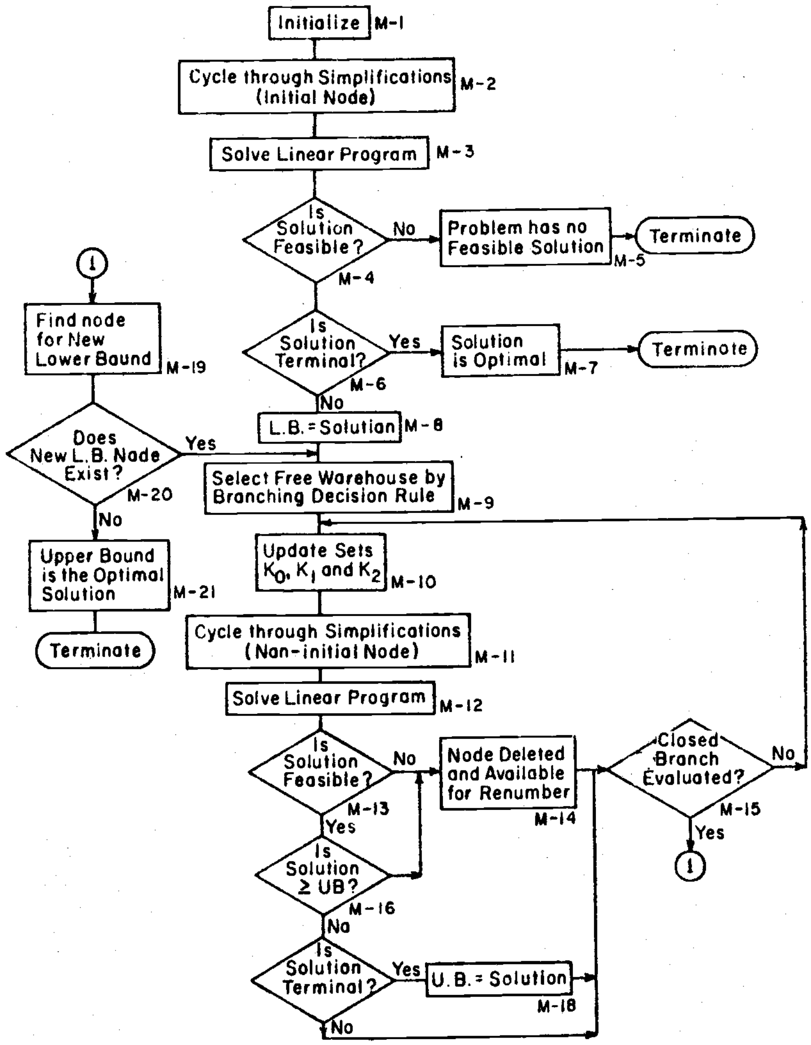


Figure 2.2[9/] Branch and bound procedure flow chart

9/ Obtained from an article by Khumawala (1972. p. B-725).

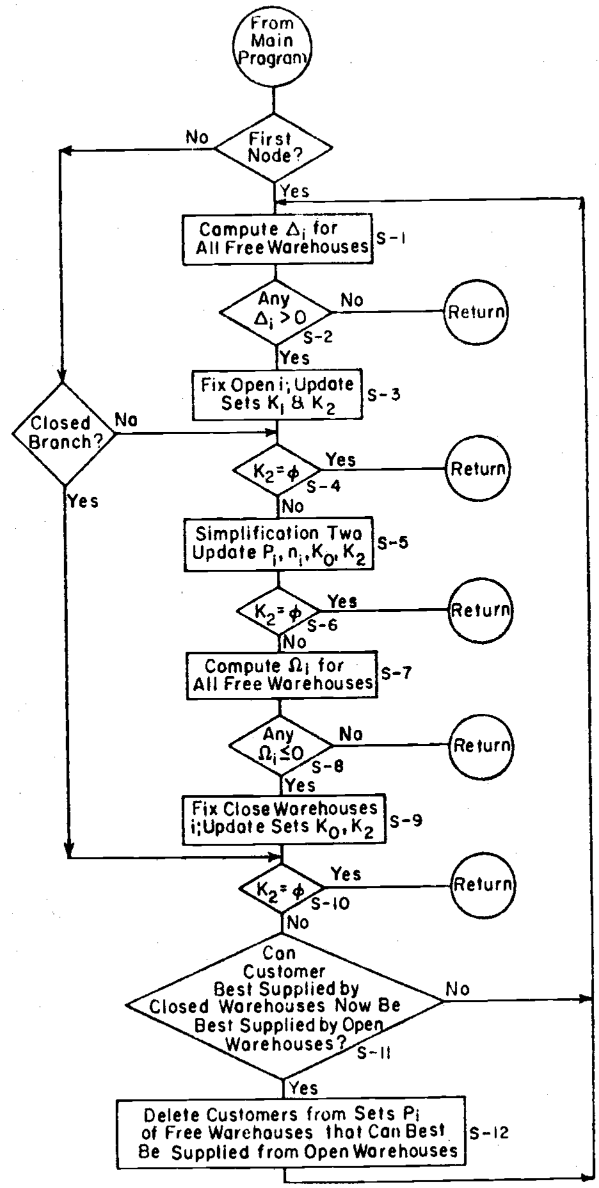


Figure 2.3[10/] Simplification cycle flow chart

10/ Obtained from an article by Khumawala (1972. p. B-724).

$$Y_i = \begin{cases} 1 & i \varepsilon K_1 \\ \dfrac{\Sigma_{j \varepsilon P_i} X_{ij}}{n_i} & i \varepsilon K_2 \\ 0 & i \varepsilon K_0 \end{cases}$$

Where: $g_k = \begin{cases} 0 & k \varepsilon K_1 \\ F_k & k \varepsilon K_2 \end{cases}$

The efficiency comes from the fact that ∇_{ij} always exists. The proof that the solution is optimal is shown in Khumawala's dissertation (1970).

In order that the branching may continue, an office must be selected from the set of free offices, K_2, at the node where further branching is to take place (step M-9). The selection of an office is done by a branching decision rule. There are several possible rules which could be used to determine the office. Khumawala experimented with some of these and found that the selection of the free office with the largest positive Ω_i was the best rule in most cases.

The selected office is first constrained opened and then constrained closed. In each case, the simplification procedures are followed. The solutions resulting are compared with the present bounds to see if they may be replaced. If the solution is terminal, it is compared with the current upper bound. If it is nonterminal, it is compared with the current lower bound. When no nonterminal nodes with solutions less than the current upper bound can be found, the procedure ends. The current upper bound is optimal. The following illustrative example explains the procedure more fully.

An Example

The following matrix shows the C_{ij} cost entries developed from the data provided in Appendix A. This is a simplified example designed to illustrate the algorithm. Each Ⓛ represents a very large cost which prohibits a city j from being serviced by office i. F_i is the opening cost. The flow charts, Figures 2.2 and 2.3, are referred to in the explanation.

TABLE II. COST MATRIX FOR DATA GIVEN IN APPENDIX A

	City j: 1	2	3	4	5	6	7	F_i
Office i 1	282	399	1020	Ⓛ	Ⓛ	Ⓛ	191	500
2	799	141	958	Ⓛ	385	579	390	500
3	Ⓛ	Ⓛ	794	71	267	738	365	500
4	Ⓛ	385	823	134	141	530	Ⓛ	500
5	Ⓛ	290	894	185	265	282	Ⓛ	500
Demand	220	110	330	55	110	220	110	

The algorithm minimizes the total cost according to equation 2.1 subject to equations 2.2 and 2.4.

The initialization M-1, involves setting $K_1 = K_0 = \emptyset$, the empty set, and $K_2 = (1,2,3,4,5)$; the sets P_i (i=1,2,3,4,5) and N_i (i=1,2,3,4,5,6,7) are also initialized. The lower bound (LB) = 0, and the upper bound (UB) = + ∞.

The simplification cycle, M-2, is entered to attempt the opening or closing of offices. In simplification one S-1, ∇_{ij} and Δ_i are computed. The values are:

$\nabla_{11} = 517$ $\quad\nabla_{17} = 174$ $\quad\Delta_1 = 192$

$\nabla_{22} = 149$ $\quad\Delta_2 = -351$

$\nabla_{33} = 29$ $\quad\nabla_{34} = 63$ $\quad\Delta_3 = -408$

$\nabla_{45} = 124$ $\quad\Delta_4 = -376$

$\nabla_{56} = 248$ $\quad\Delta_5 = -252$

It is found from this simplification that office number 1 should be opened (S-3), $Y_1 = 1$, since $\Delta_1 > 0$. In other words, $K_1 = \{1\}$, $K_2 = \{2,3,4,5\}$, $K_0 = \emptyset$. It pays to open the office because it is more expensive to make people go elsewhere. Sets P_i and n_i are updated in the second simplification, S-5. Since demand centers 1 and 7 are best serviced by office 1, they are eliminated from further consideration as customers for the other offices. The omega values are calculated in simplification three (S-7). No $\Omega_i \leq 0$ so no offices can be closed. The procedure returns to the main program because of this (S-8).

The linear program, M-3, is now solved. It is best that customers 1 and 7 go to office 1 since ∇_{17} and ∇_{11} are positive. These are elements of S_1. Therefore, $X_{11} = 1$ and $X_{17} = 1$. X_{22}, X_{33}, and X_{34} are set equal to one because demand centers 2, 3, and 4 of $S_1 \cup S_2$ are best serviced by offices 2, 3, and 3 respectively. Finally, X_{45} and X_{56} equal one because $\nabla_{45} \geq F_4/n_4$ and $\nabla_{56} \geq F_5/n_5$

respectively. They are elements of S_2. All other $X_{ij} = 0$. $Y_1 = 1$ because $K_1 = 1$, and $Y_2 = 0.2$, $Y_3 = 0.4$, $Y_4 = 0.2$, $Y_5 = 0.2$ because $K_2 = \{2,3,4,5\}$. The solution to the LP is $Ƶ = 2901$. It is feasible and nonterminal. 2901 becomes the lower bound (LB at step M-8). If the solution had been terminal, the procedure would have terminated.

The procedure continues at step M-9 where a free office ($K_2 = \{2,3,4,5\}$)is selected by the branching decision rule. This office is first opened and then it is closed. Office number 3 is selected as the office on which to branch. The program enters the simplification cycle at S-4. As a result of simplification three, offices 2 and 4 are closed. ($\Omega_2 \leq 0$ and $\Omega_4 \leq 0$). The procedure goes back to the beginning of the simplification cycle (S-1). Office 5 is opened because of simplification one ($\nabla_5 \geq 0$). The procedure returns to the main program (M-12) because $K_2 = \emptyset$. The LP solution is:

$$X_{11} = X_{52} = X_{33} = X_{34} = X_{55} = X_{56} = X_{17} = 1 \ (\nabla_{ij} \geq 0),$$

$$Y_1 = 1,\ Y_2 = 0,\ Y_3 = 1,\ Y_4 = 0\ ,\ Y_5 = 1, \text{and } Ƶ = 3484.$$

This is a terminal solution and it becomes the new upper bound (UB) at step M-18.

The procedure continues by closing office 3 (M-15). The simplification cycle is entered but no offices are opened or closed. The LP is solved. The resulting solution is: $Ƶ = 2993$. It is nonterminal. It is the only nonterminal node left and it has a solution which is less than the current upper bound, 3673. Therefore, it is branched on next.

Office 4 is picked as the next office on which to branch. The procedure continues much the same as the preceeding portion. The branch and bound tree (Figure 3.3) shows the results of the remainder of the program. The program terminates because there are no more nonterminal nodes to branch onto next. The optimal solution becomes the minimum of the values at the terminal nodes. It is 3386. Offices opened and the cities serviced by them are:

Office 1 services demand centers 1 and 7, and

Office 5 services demand centers 2,3,4,5, and 6.

In Figure 3.3 one of the efficiencies used to minimize storage needs for the algorithm is shown. All of the information contained in the node marked with an X, node 3, is no longer needed for algorithm after the branching decision is made. Thus, instead of numbering the branch nodes 4 and 5, they are numbered 3 and 4. The procedure is effective for large problems.

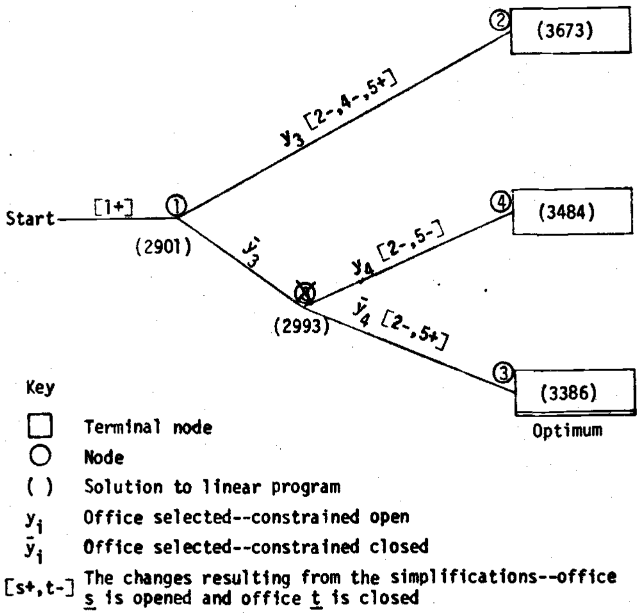


Figure 2.4. Branch and bound tree for the example

III. THE COLLECTION OF DATA

Data collection is probably one of the most critical parts of any study. The collection of the data in this study was simplified by the cooperation of the DMV's Director of Field Services, Harvey Ward. Since inaccurate data obviously will result in erroneous results, it is vital that accurate and relevant figures are selected. The data must also fit the requirements of the model which requires that the unit cost, γ_{ij}, the demands, D_j, and the opening cost, F_i, be defined. To be consistent, all data will pertain to the year 1972.

The people of Oregon must pay for the operation of the DMV through taxes. They must also pay for the expense incurred while traveling to the field offices. Therefore, it is reasonable to minimize the total cost to the public, the object of the formulation. Referring to the objective function, Equation 2.1, there are two costs which must be evaluated:

1) $C_{ij} = \gamma_{ij}D_j$ is the cost matrix associated with the demand centers and the candidate field offices.

2) F_i is the cost of opening a field office.

Some representation for demand is needed in order to evaluate the needs of each demand center and the cost matrix C_{ij}.

The demand was probably the most difficult to determine. It is logical to assume that the demand centers are the cities in the state. Those people living in the rural areas are included in the city closest to their home. Ideally, by knowing the number of trips made from each demand center to the present field offices to make transactions, the needs of the people can be evaluated. This information is not available. Therefore, some other data which represents demand must be used. It was suggested that the population census be used to represent the demand. A report was obtained from Portland State University showing the population of each of the incorporated cities and the population of the unincorporated cities and the population of the unincorporated areas by counties. The population of the unincorporated areas is quite substantial, but there is no way of determining where these people live without going back to the census track data. If the population data were used, then some factor for converting the population into representative demand would be needed. While investigating the use of population, a much better representation of demand was found.

Why not use the data which the DMV has on master file? The mailing addresses of all the drivers of record are known. Thus, one can list all the Zip Codes (cities) and the number of drivers of record at each Zip Code.[11/] The only problem with this data was its availability. At the time it was originally requested, it was not available; but it became available later. Using this information, the demand is represented as a proportion of the number of drivers of record in each demand center, DR_j (j=1,2,...n). It is assumed that each driver represents 1.10 transactions per year.[12/] It will be assumed that each transaction

---

11/ This information is shown in Appendix D. The reason for using Zip Codes is to make it possible to divide the State into smaller sections which may be necessary because of the storage limitations of the computer. The use of Zip Codes has merit because the boundaries follow roads and natural barriers. In the end, it was necessary to divide the state into four parts.

12/ The calculations are shown in Appendix C.

represents one trip to the field office. This is not totally true because some drivers have two vehicles and may make two transactions in one trip, but this is still the best measure of the number of trips that are made. The demand (trips or transactions) is represented by:

$$D_j = DR_j \cdot (1.10).$$

Each of the demand centers could be used as a possible field office location. But, this is neither logical management-wise, nor is it reasonable when considering the storage capacity of a computer. In reality, the DMV would not consider locating an office in a very small town. Accordingly, we decided that any town with a driver population less than 2000 people would not be considered as a candidate. This constraint reduced the number of candidate offices to 114 as shown in Appendix D.

Now that there is a representation for demand, costs must be determined. The unit cost, γ_{ij}, is composed of two main parts: 1) the cost to the public for the travel from the demand center j to the candidate office i; and 2) the operating costs of the field offices. To determine the cost to the public, the distances between the demand centers and field offices must be evaluated. They can either be represented by the actual miles or a mathematical representation. From a practical standpoint, the mathematical representation is better because the determination of the mileage is much easier, and the storage of the data is not as large a problem. With a problem with 114 possible offices and 417 demand centers, a large matrix would have to be stored if the actual distances were used. Another reason for using the mathematical representation is the ease of making changes in the data set. For example, if the problem needs to be reduced in size, the amount of data that must be manipulated is much smaller. One reservation is that it is not as accurate as the actual data, but it gives a close representation. For this problem the distance is given by

$$\text{Miles}_{ij} = \sqrt{(Z_{1i} - X_j)^2 + (Z_{2i} - Y_j)^2} \cdot (\text{Scale})$$

where (Z_{1i}, Z_{2i}) is the office location, (X_j, Y_j) is the demand center location,[13/] and Scale is the number of miles per unit of measure (1.875 miles per unit). Finding the coordinates of each city or demand center involved the plotting of the cities on a grid. This was quite a lengthy process, but was much easier than finding the actual distances between the cities.

The distance to a field office is used as a screening device. If it is necessary to travel a long distance to a field office, then a very large cost is associated with the route. It works in the same manner as the L cost in the example problem. The Director of Field Services requested that:

1) The people in Eastern Oregon not travel more than 150 miles one way.
2) The people in Western Oregon not travel more than 50 miles one way.
3) Those in the Portland Metropolitan area not travel more than 10 miles one way.

13/ The coordinate locations are shown in Appendix D.

The 10 mile constraint was not used and was not necessary because the cost of travel constrained the distance traveled in the Metropolitan area to be less than 10 miles.

The cost per unit demand for travel is given by:

$$(\text{Miles}_{ij}) \cdot (\text{Rate}) \cdot 2.$$

The rate should include the cost of travel and the cost of inconvenience to the public. The cost of travel is set at 10¢ per mile per trip since this is the amount that the State allows for its travel. The cost of the public's time is set at $2.00 per hour because this is approximately the minimum wage. The cost of inconvenience is a hard factor to evaluate. For some people the inconvenience is great, yet for others it is minimal. For this study, the cost of inconvenience will be included in the $2.00 per hour allotted for the public's time. This value seems reasonable because some of the people coming to the field offices for licenses or vehicle registrations have no income, some are on welfare, some make the minimum wage, and some, or course, have large salaries or wages. Also, some combine the trip to the field office with other errands and thereby lessen the cost of inconvenience. Therefore, it will be assumed that on the average the cost of inconvenience is included in the $2.00. If it is assumed that people overall average 25 miles per hour going to the field office, making the transaction, and going home, then an estimate of the cost of inconvenience can be made in cost per mile.

$$\$2.00 \text{ per hour}/25 \text{ mph} = \$.08 \text{ per mile}$$

Combining this cost of inconvenience with the $.10 per mile, an estimated cost of \$.18 per mile results. The distance between office i and demand center j does not represent a round trip. Therefore, either the cost per mile or the distance must be doubled before they can be used to calculate the C_{ij} entries.

The total cost of travel function is shown in Figure 3.1. As the number of offices increases, the cost of travel decreases. It will be assumed that no travel cost is associated with a field office located in a demand center. In the analysis, the traveling cost is varied from 10¢ to 14¢ to 18¢ per mile.

Also, included in γ_{ij} is the cost of operating the field offices. The amount budgeted for 1972 is used in the calculations. About \$10,120[14/] was budgeted per employee which is about \$1.41 per driver of record. Normally, the cost of operations increases with greater decentralization because of increased administrative costs such as supervisory and communication costs. (Line 3 Figure 3.1) An increase in rent and maintenance, resulting from the need of more office space will also alter the cost of operations. It is assumed for this problem that the total number of employees[15/] needed to man the offices remains constant and operating cost does not increase with an increase in the number of offices opened. C_{ij} becomes:

14/ The calculations are shown in Appendix C.

15/ 218.75 employees are needed to maintain the services for the Vehicle Registration and Drivers license business. Calculations shown in Appendix C.

$$[(\text{Miles}_{ij}) \cdot \$.18 \cdot 2 + \$1.28]\ D_j$$

Where: $1.28 = $1.41/1.10 = the cost per transaction

The cost of opening an office, F_i, was initially set at $20,240. The Director of Field Services wants at least enough work for a two-man office before he would open it. The $20,240 is the operating cost for an average 2-man office for one year. This opening cost does not guarantee that each office will have two employees; it only guarantees it is worth spending $20,240 to open the office. This cost seems low, so opening costs of $30,240 and $40,240 are also used to test the sensitivity of the results. The opening cost is the same for each office, although it could have varied with the offices. For example, the cost of opening one of the existing offices could be assigned a zero cost while the opening of nonexisting offices could be assigned a large cost. For this problem, the desire is to find out where the offices should be located without considering the present locations. Therefore, it is assumed that there are no existing offices. The cost, F_i, shown in Figure 3.1 is a step function (line 2). The total cost curve is also a step function because of F_i.

The complete objective function can now be given. It is:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} [(\text{Miles}_{ij}) \cdot (\text{Rate})\ 2 + \$1.28]\ D_j\ X_{ij} + \sum_{i=1}^{m} F_i Y_i$$

where

Miles_{ij} - is the number of miles from office i to demand center j.

Rate - is the cost per mile with values of $0.10, $0.14 and $0.18.

D_j - is the demand in transactions at center j

F_i - is the fixed cost with values of \$20,240, \$30,240 and \$40,240.

The cost functions and their interactions is shown in Figure 3.1. As the number of offices opened increases, the direct costs (the travel costs) to the public decreases and the indirect costs (the costs of opening and operating the field offices) increases.

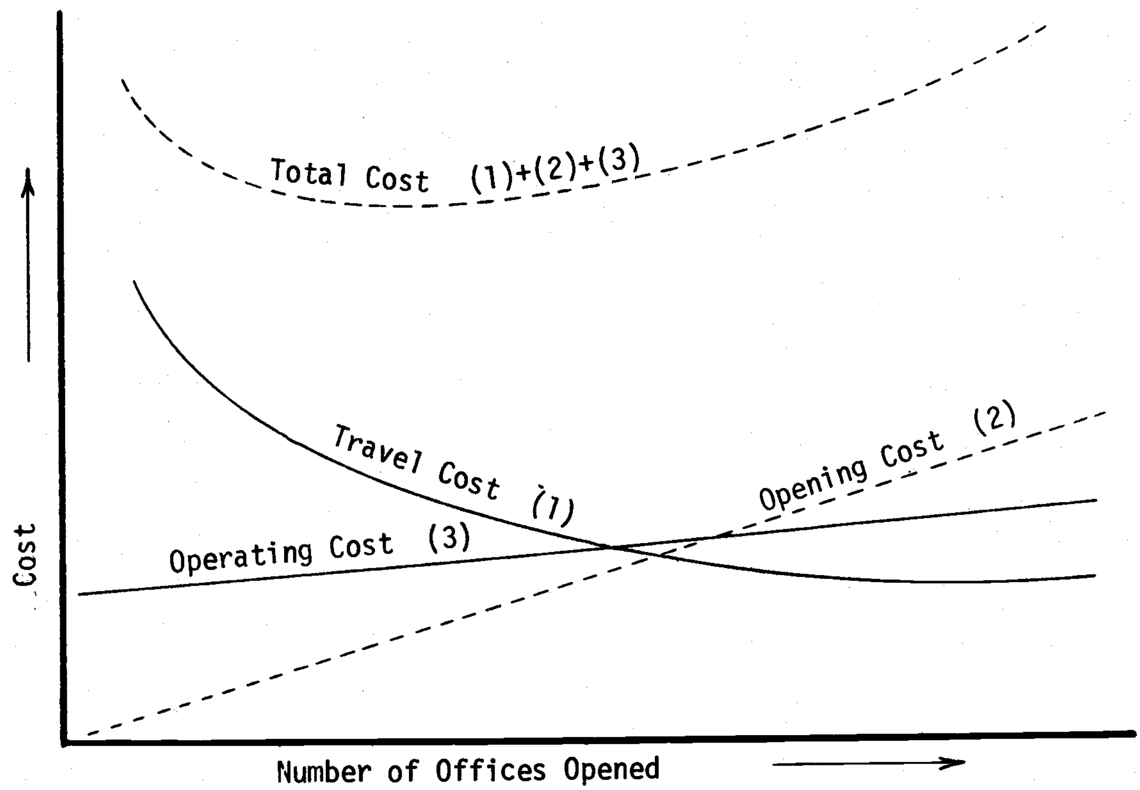


Figure 3.1 Relationships of Costs

Now that there are representations for demand, distances, and cost, the computer runs can be made to determine the location of the offices.

IV. THE ANALYSIS

The analysis of the data involved the making of several computer runs[16]. A sensitivity analysis was performed to investigate the effects of changes in the opening and travel costs on the offices opened (number and location), the staffing requirements, and the total cost. These changes also affect the difficulty of determining the optimal solution. The difficulty is shown by the number of nodes that must be investigated or by the size of the branch and bound tree and by the amount of computer time used. For this problem, only the effects on the number of nodes are investigated. The number of nodes used is directly related to the amount of computer time.

The opening cost was varied from \$20,240 to \$30,240 to \$40,240; the travel cost was varied from \$0.10 per mile to \$0.14 per mile to \$0.18 per mile.

Initially, an attempt was made to solve the office location problem by making one large run which included all 417 demand centers and 114 candidate offices. Because the storage capacity of the computer was not large enough, the problem was broken down into four parts. The four areas are shown in Figure (4.4a). The use of Zip Codes for the break-down was quite effective. The break between 2 and 4, and 3 and 4 follows natural barriers and as a result has very little affect on the solution. On the other hand,

---

[16] The runs were made on an IBM 370-158 used by the Motor Vehicles Division.

the division between 1, 2 and 3 may have some affect on the offices opened near the border. Offices to which the demand centers are assigned seem to be affected more than the actual offices opened. But, there is no proof since the groups were not combined.

Areas 1 and 2 were initially together, but there was not enough storage to find a solution. The limit set on the number of nodes which could be investigated was 61. Even by breaking the problem down into two smaller problems and increasing the number of possible nodes to investigate to 151, a solution[17/] could not be found in some cases. In others, optimality could not be ascertained. In cases where a solution was found but not determined to be optimal, the computer code printed, "The solution given below may not be optimal because of lack of storage."

A large amount of computational experience was obtained during the analysis of the data. One point of interest is the results obtained from a run in which an error was made. It occurred at line 272 in the computer code (Appendix E). Instead of having:

```
XX = 1./XLN,
XX = IFC(KW)/XLN
```

was in its place. As a result of this error, the wrong lower bounds for the non-integer solutions were calculated (much larger than the correct values). This decreased the computational difficulty in finding a terminal solution because very few nonterminal solutions were stored. The procedure ended promptly when a terminal solution

---

17/ A solution meets the integer constraints, but is not necessarily optimal.

was found because no nodes could be found with a lower bound less than that solution. Hence, it was not determined to be optimal. However, the difference between the total cost in the modified branch and bound (computer code with the error) and the solutions obtained using the regular procedure averaged 0.35%. The difference ranged from no error to an error of 3.13%.[18/] The results are shown in Table III. The number of nodes which had to be investigated by the modified branch and bound, was much less in most cases. On an overall average, the modified procedure took 43 fewer nodes to solve the problem. The regular procedure averaged 50 nodes in determining optimality and the modified procedure averaged 6.8 nodes in finding a solution.[19/] This average should be somewhat larger because in some cases no solution could be found for the regular procedure. Since the computational difficulty is so much less and the solutions near optimal, the modified procedure could be used as a heuristic type of method to find "good" solutions for large problems.

Changes in the unit cost per mile and the opening cost per office affect the number of nodes which must be used to determine the optimal solution. The difficulty of determining an optimal solution is also affected by the density of the demand centers.

18/ The maximum error may be greater than is shown because in this case there was no way of determining if the solutions were optimal.

19/ The original data is shown in Appendix B.

TABLE III. DIFFERENCES BETWEEN THE TOTAL COSTS OBTAINED ON THE VARIOUS RUNS.

The Runs			Modified B&B	Regular B&B	Error in Total Cost
Travel Cost	Opening Cost	Area			
$.10	$20,240	#1	$1,158,996.00	$1,156,885.00	.25%
		#2	656,192.00	654,496.50	.18%
		#3	1,450,871.00	1,450,871.00	No Error
		#4	1,116,210.00	1,105,804.00	.94%
	$30,240	#1	$1,316,331.00	No Solution	---
		#2	762,036.75	(740,402.81)*	2.9%
		#3	1,651,470.00	No Solution	---
		#4	1,285,421.00	1,285,421.00	No Error
	$40,240	#1	$1,397,209.00	No Solution	---
		#2	821,193.00	(827,912.19)	.81%
		#3	1,779,241.00	No Solution	---
		#4	1,437,594.00	1,437,594.00	No Error
$.14	$20,240	#1	$1,190,242.00	$1,190,242.00	No Error
		#2	707,627.37	707,627.37	No Error
		#3	1,552,954.00	1,552,954.00	No Error
		#4	1,222,064.00	1,222,064.00	No Error
	$30,240	#1	$1,346,435.00	$1,336,416.00	.7%
		#2	814,425.44	813,954.87	.05%
		#3	1,786,718.00	1,786,718.00	No Error
		#4	1,418,778.00	1,418,778.00	No Error

* The total costs given in parentheses are not necessarily optimal.

TABLE III. (cont.)

The Runs			Modified B&B	Regular B&B	Error in Total Cost
Travel Cost	Opening Cost	Area			
$.14	$40,240	#1	$1,454,315.00	$1,451,360.00	.20%
		#2	928,703.06	(900,511.00)	3.13%
		#3	1,991,399.00	No Solution	---
		#4	1,599,141.00	1,599,141.00	No Error
$.18	$20,240	#1	$1,213,112.00	$1,213,112.00	No Error
		#2	759,448.25	759,109.94	.04%
		#3	1,651,055.00	1,651,055.00	No Error
		#4	1,331,554.00	1,331,554.00	No Error
	$30,240	#1	$1,404,776.00	$1,381,000.00	1.17%
		#2	879,238.31	869,534.56	1.12%
		#3	1,893,206.00	No Solution	---
		#4	1,538,440.00	1,538,440.00	No Error
	$40,240	#1	$1,527,199.00	$1,515,679.00	.76%
		#2	972,651.87	972,651.87	No Error
		#3	2,122,472.00	2,122,472.00	No Error
		#4	1,731,751.00	1,731,751.00	No Error

The total error between the two procedures is 10.63%.

The average error is .35% per problem.

The analysis of variance in Table IV shows that the means given in Table V for all of the conditions expressed above are significantly different. The F test is significant for all of the conditions at the 90th percentile or higher. By observing Table V, three general statements can be made about the results within the limits of the study:

1. It is much easier to find a solution in Eastern Oregon (area 4) than it is in Western Oregon (area 1, 2, and 3). The number of nodes used is affected to a certain extent by the idiocyncracies of the problem, but a major portion of the difficulty appears to result from the density of the demand areas.
2. The difficulty of finding an optimal solution decreases as the cost per mile (travel cost) increases. The number of nodes used will reach a minimum at some cost, but no further conclusions can be made without further study.
3. The difficulty of finding an optimal solution increases as the opening cost increases. The number of nodes used will reach a maximum at some opening cost, but further study is needed to determine this cost.

It is the relationship between the opening cost and travel cost that affects the difficulty. A change in the travel cost, which is seen in the cost matrix, affects the magnitude of the costs savings for a demand center that results if a specific office is opened.

TABLE IV. THREE FACTOR ANALYSIS OF VARIANCE -- REGULAR PROCEDURE[20/]

Source	d.f.	SS	MS	F
Area	3	41957.8611	13985.9537	8.9628***
Cost per mile	2	11762.0000	5881.0000	3.7688**
Opening cost	2	9438.5000	4719.2500	3.024*
Error	28	43692.3889	1560.4424	
Total	35	106850.7500		

*** Significant at the 99th percentile F(3,24)
** Significant at the 95th percentile F(2,24)
* Significant at the 90th percentile F(2,24)

TABLE V. THE MEAN NUMBER OF NODES

Means

Area	1	2	3	4
# of nodes	55.2222+	99.5556+	41.4444+	4.1111

Cost per mile	$.10	$.14	$.18
# of nodes	74.9167+	42.9167+	32.4167+

Opening cost	$20,240	$30,240	$40,240
# of nodes	27,2500	60.0000+	63.0000+

+These means should be somewhat higher because for some of the problems the storage limit was reached. Therefore, the actual number of nodes that it would take to find a solution is not shown.

20/ The original data are found in Appendix B.

As this cost savings decreases in relation to the opening cost, it becomes much more difficult for the algorithm to determine which offices to open. A larger search must be made to investigate the opening of offices because fewer offices are opened or closed by the simplified procedures; they must be opened by the branching decision rule.

Not only do changes in the travel cost and opening cost affect the difficulty of computation, but they also affect the solutions. The effect of the total cost is shown in Figure 4.1. It is an increasing function because an increase in travel costs or opening costs must be reflected as an increase in the total cost. The total cost is a representation of the miles traveled by the public and the offices opened.

Looking at Figure 4.2, it can be seen how the changes in travel cost and opening cost affect the total miles that the public travels. The number miles traveled is inversely proportional to the cost per mile and directly proportional to the opening cost. As the cost of travel decreases, people can afford to travel farther. If Figure 4.3 is looked at along with Figure 4.2, a better picture is obtained. At a fixed opening cost with decreasing travel cost, fewer offices have to be opened because people can afford to travel farther. On the other hand, at a fixed travel cost with increasing opening cost, the public is forced to travel farther because fewer offices are opened. The range of offices opened varies from a maximum of 77 at $0.18 and $20,240 to a minimum of 42 at $0.10 and $40,240.

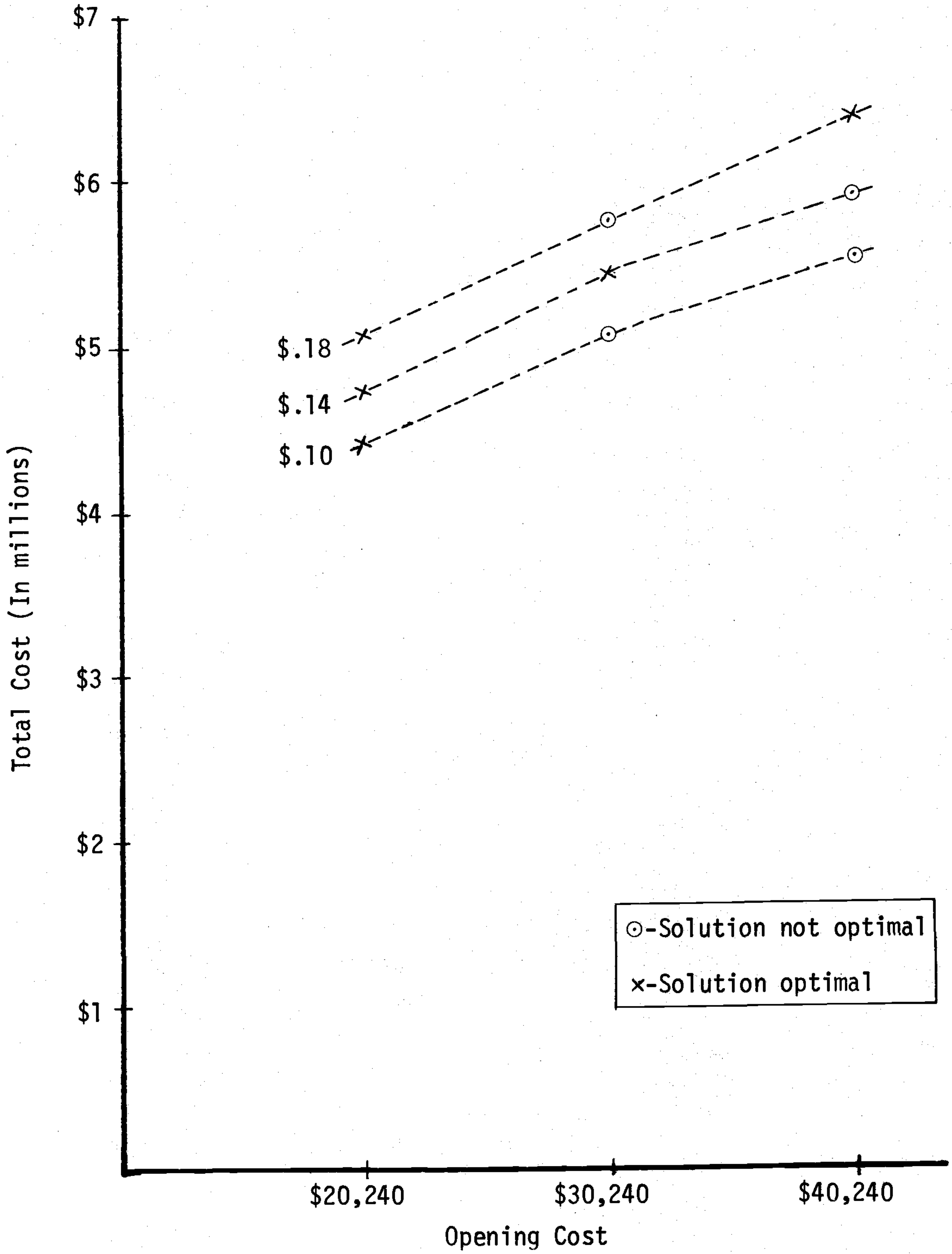


Figure 4.1. Effect on the total cost.

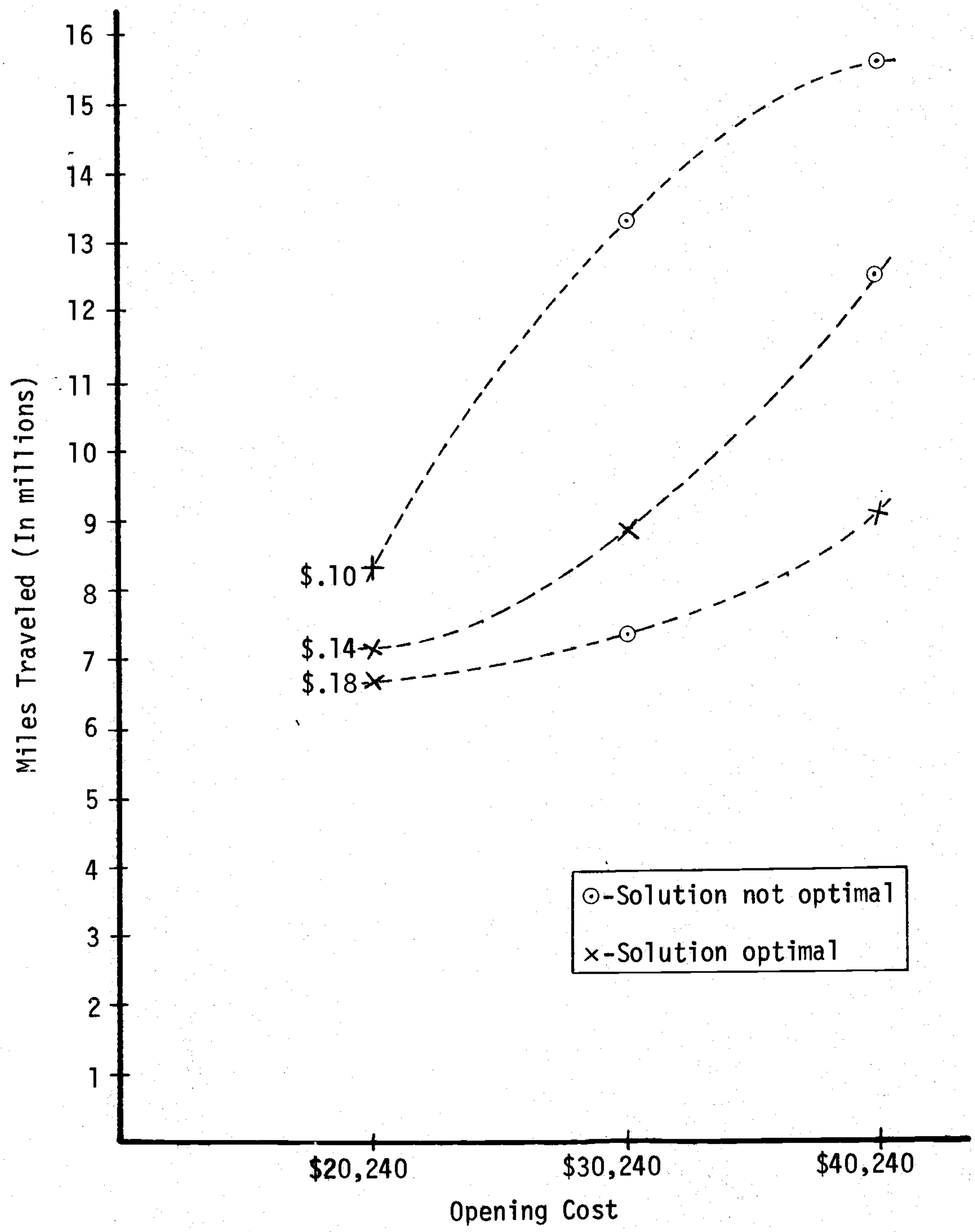


Figure 4.2. Effect on the total miles traveled.

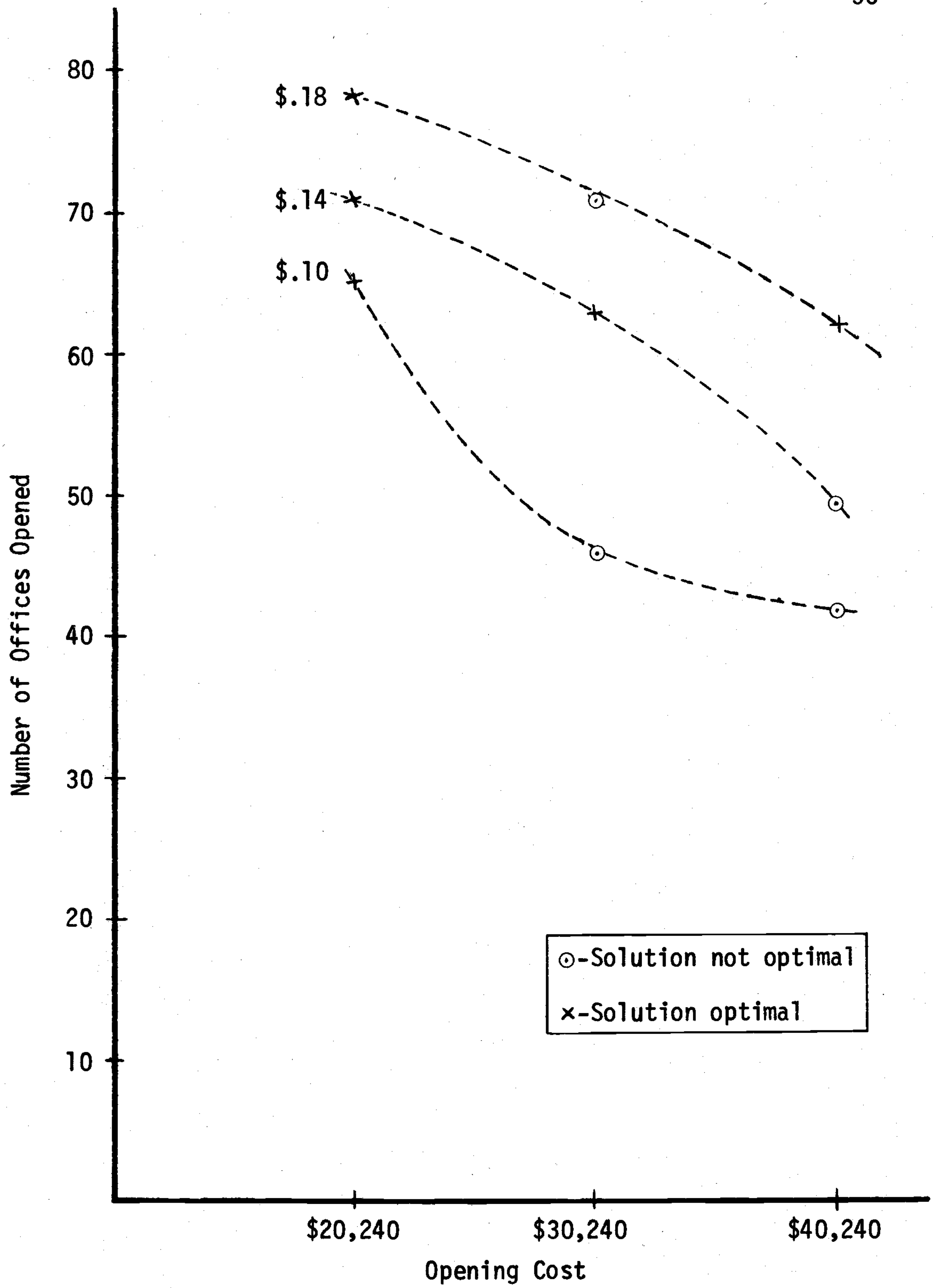


Figure 4.3. Effects on the number of offices opened.

In both of the graphs, the lines should intersect when the opening cost is zero. All possible offices are opened at this point no matter what the travel cost is. In Figure 4.2 and 4.3, it appears that the lines will intersect before they reach zero opening cost. This is possible because the lines behave like step functions. The dotted lines between the points represent the general direction of increase or decrease not the actual functions. From zero opening cost to some greater cost, X, all of the offices will remain open. The people will continue to travel the same distance as long as the same number offices are open. From cost X to another greater cost Y, one less office is open. These steps continue until the minimum number of offices are opened. The lines must also intersect at the other end at some opening cost M where the fewest number of offices can be open. The minimum is limited by the number of prohibited routes in the C_{ij} matrix. If there were no prohibited routes, only one office would open. Since the curves intersect at both ends, the concave nature of the top line and the convex nature of the bottom line are reasonable.

Finally, the actual locations determined by the model are affected by changes in the costs. On the following maps, all of the possible office locations are represented by circles and squares. The squares represent the locations where the DMV presently has its field offices located. The circles are locations with driver populations of at least 2000 people. The locations seem to be very reasonable because they coincide greatly with the present locations.

In fact, Figure 4.6a shows the results of a run in which the algorithm opened almost the same offices which are open now.

The locations given here can be used to help determine where the field offices should be located. They should not be used as the absolute answers. There are many assumptions made to make it possible to be solved on a computer. Therefore, if the DMV uses the solutions determined by the model to locate new offices, they should evaluate the peculiar needs of each area before making a final decision.

Some general comments can be made about the results. The runs made at a travel cost of $0.18 per mile will be given the most attention because the costs are more realistic.

1. Jordan Valley and Umatilla offices which are presently open were never opened by the algorithm. The DMV uses these offices for handling only Public Utility Commission and Highway transaction business. Since the model locates the field offices according to Vehicle Registration and Driver License business, the results agree with the fact that no Registration or License business is handled in these offices.
2. In Area 4, offices in Enterprise, Redmond, Madras, and Talent are consistently opened. There are presently no offices in these towns, therefore, it is recommended that these be considered in the future.
3. Portland appears to need more offices than are presently

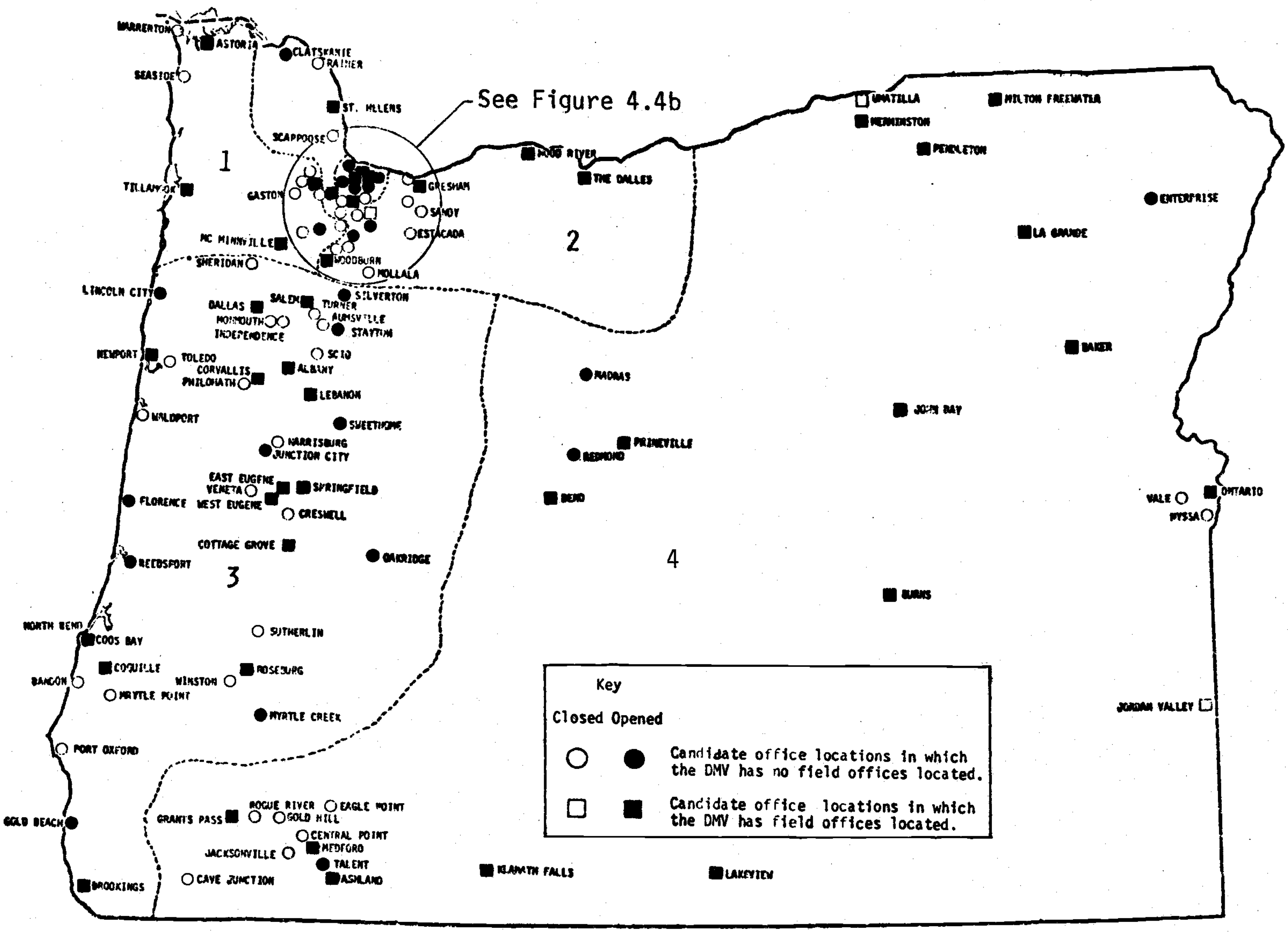


Figure 4.4a. Office locations from runs with an opening cost of $20,240 and a travel cost of $.10 per mile.

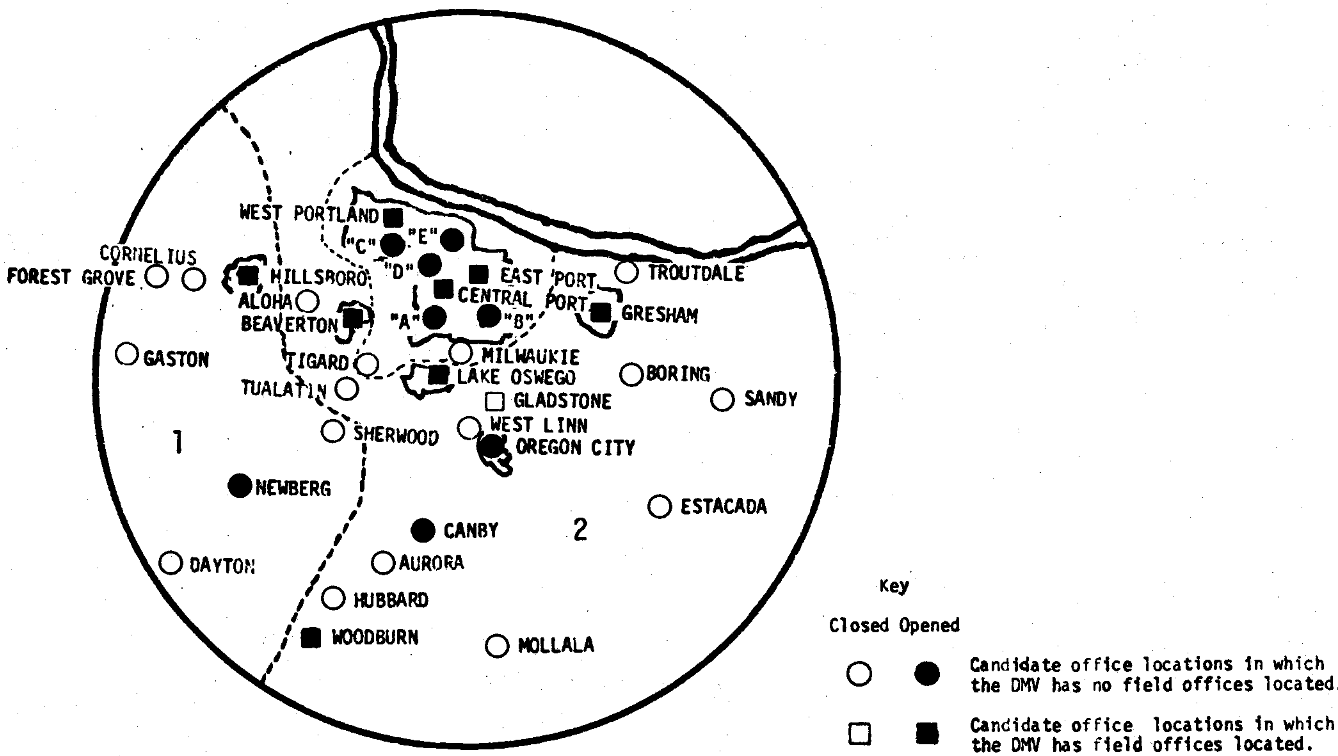


Figure 4.4b. Enlargement of Portland and the surrounding area.

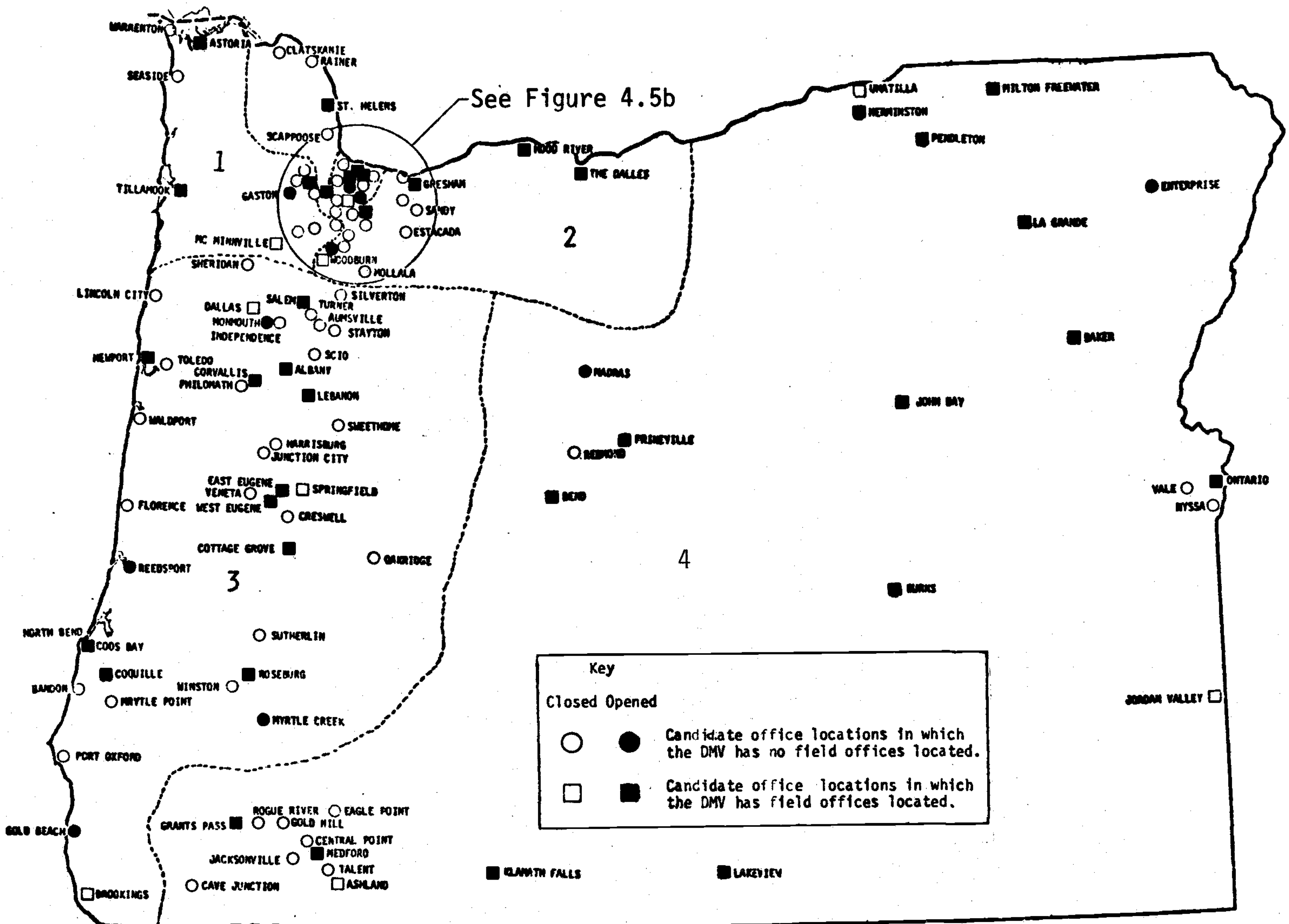


Figure 4.5a. Office locations from runs with an opening cost of $30,240 and a travel cost of $.10 per mile.

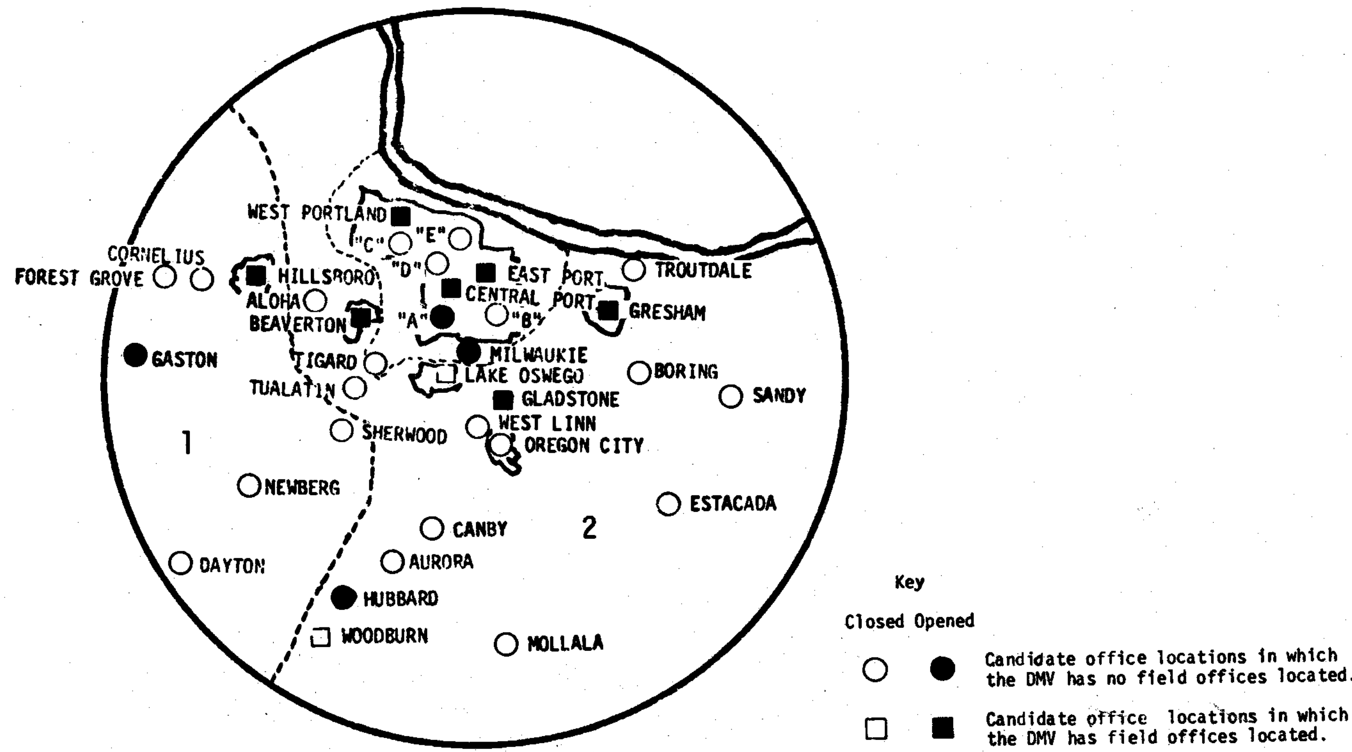


Figure 4.5b. Enlargement of Portland and the surrounding area

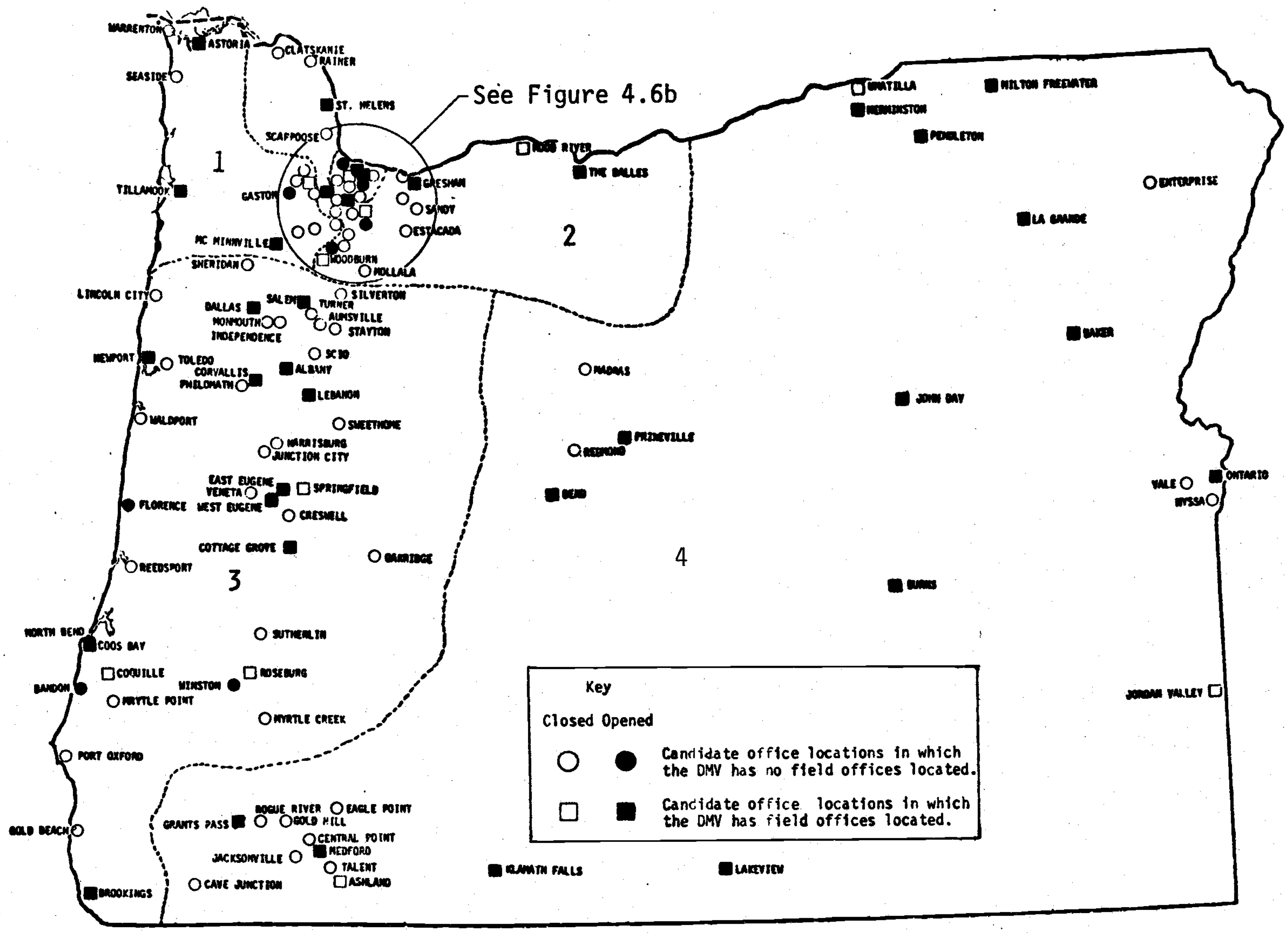
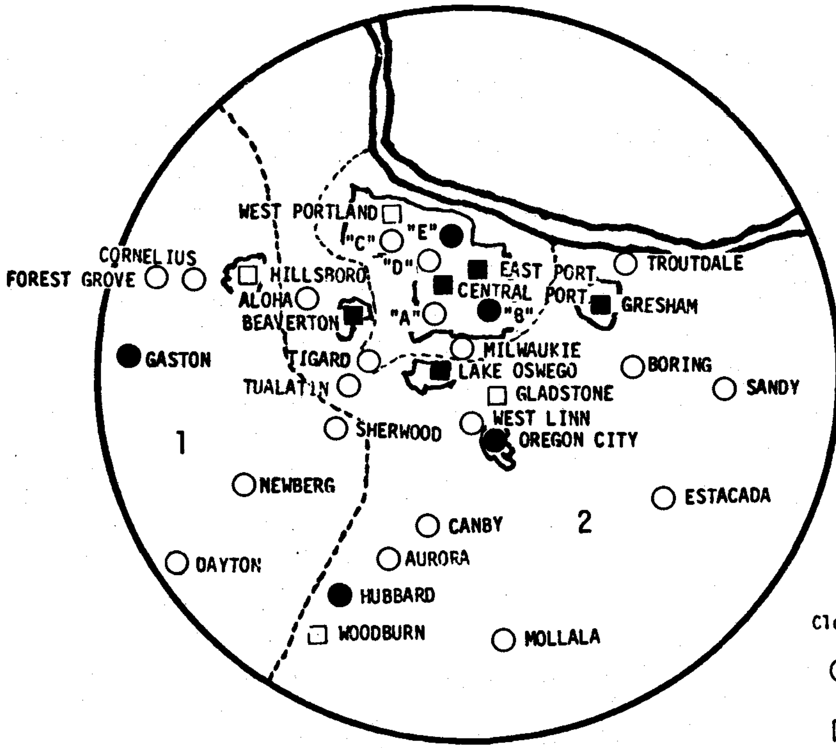


Figure 4.6a. Office locations from runs with an opening cost of $40,240 and a travel cost of $.10 per mile.



Key

Closed Opened

○ ● Candidate office locations in which the DMV has no field offices located.

□ ■ Candidate office locations in which the DMV has field offices located.

Figure 4.6b. Enlargement of Portland and the surrounding area.

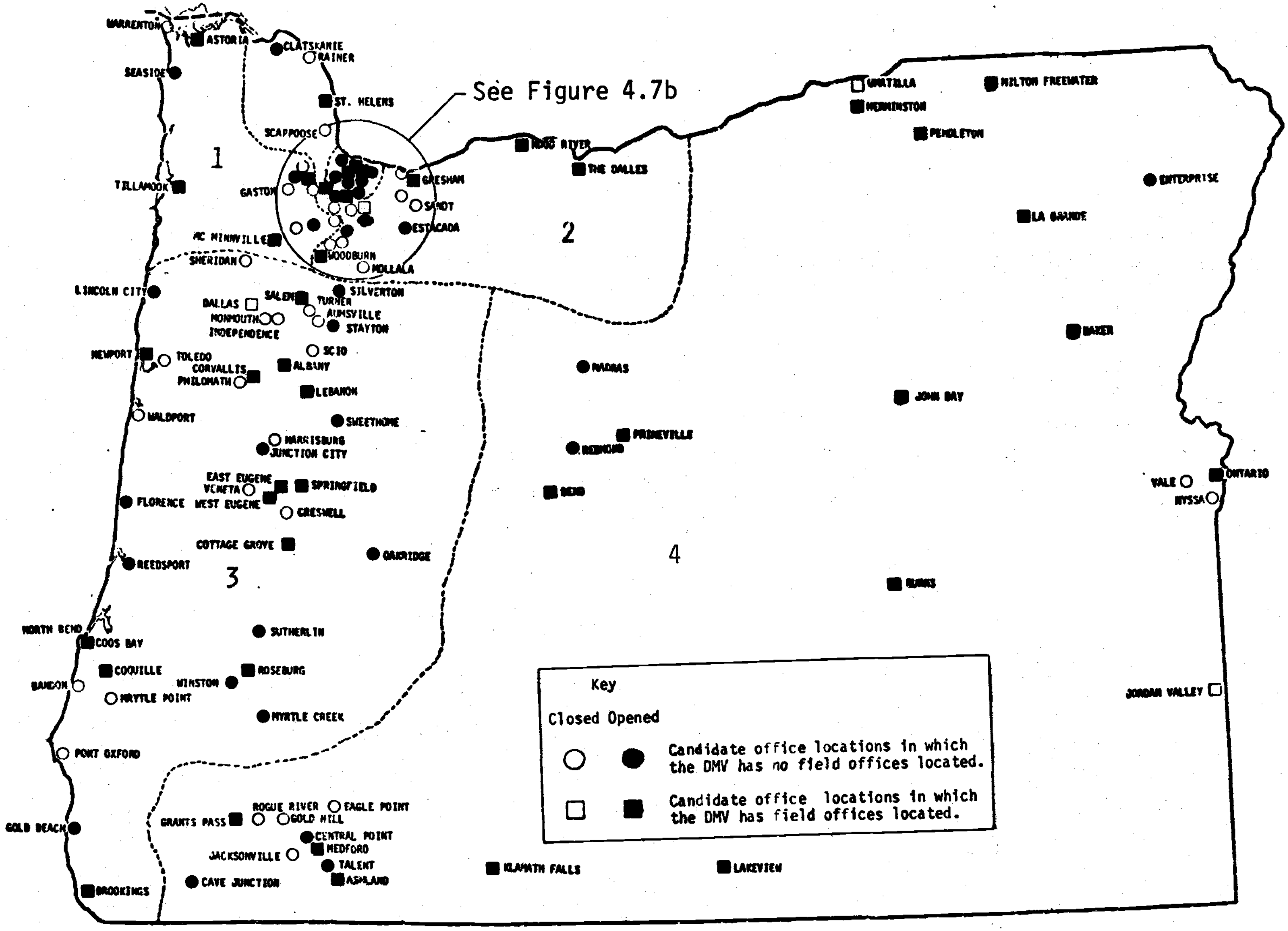


Figure 4.7a. Office locations from runs with an opening cost of $20,240 and a travel cost of $.14 per mile.

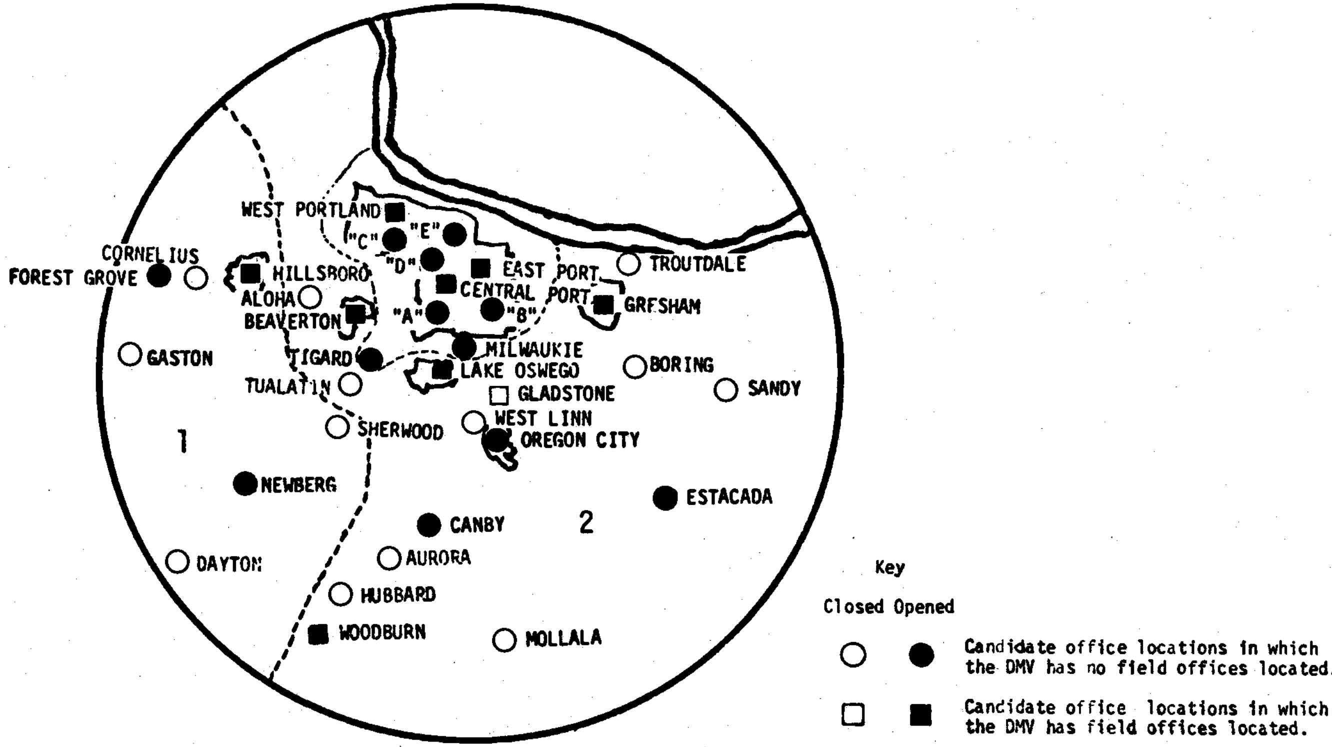


Figure 4.7b. Enlargement of Portland and the surrounding area.

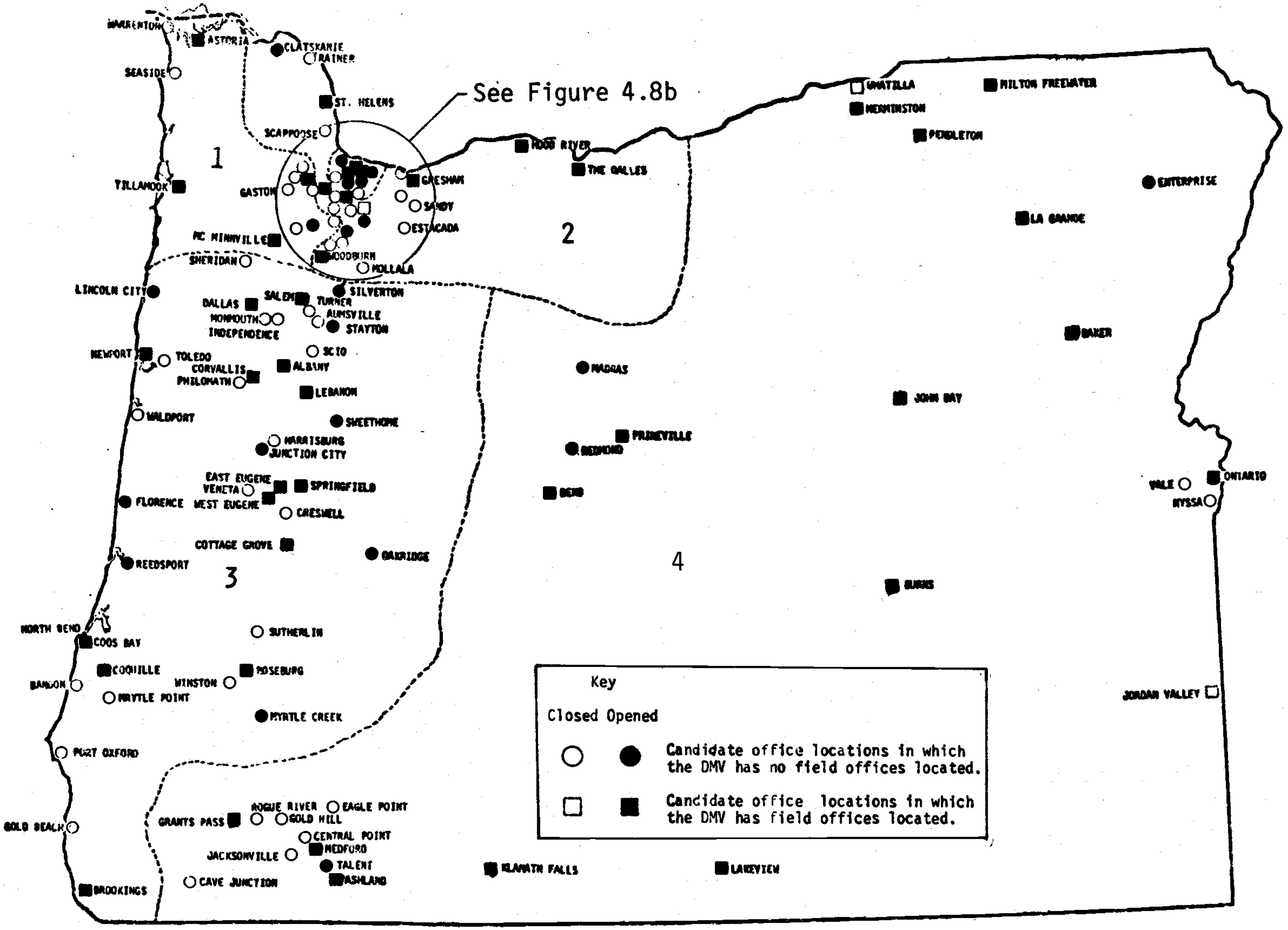


Figure 4.8a. Office locations from runs with an opening cost of $30,240 and a travel cost of $.14 per mile.

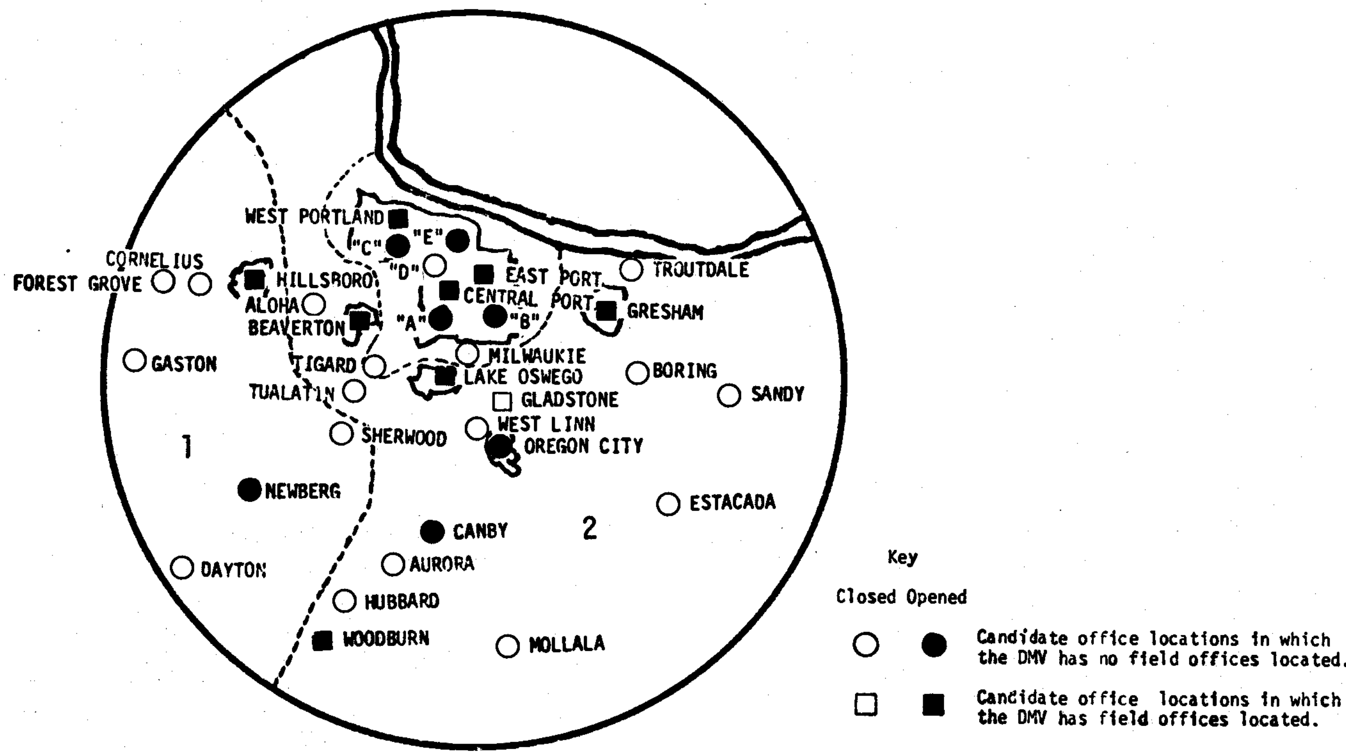


Figure 4.8b. Enlargement of Portland and the surrounding area.

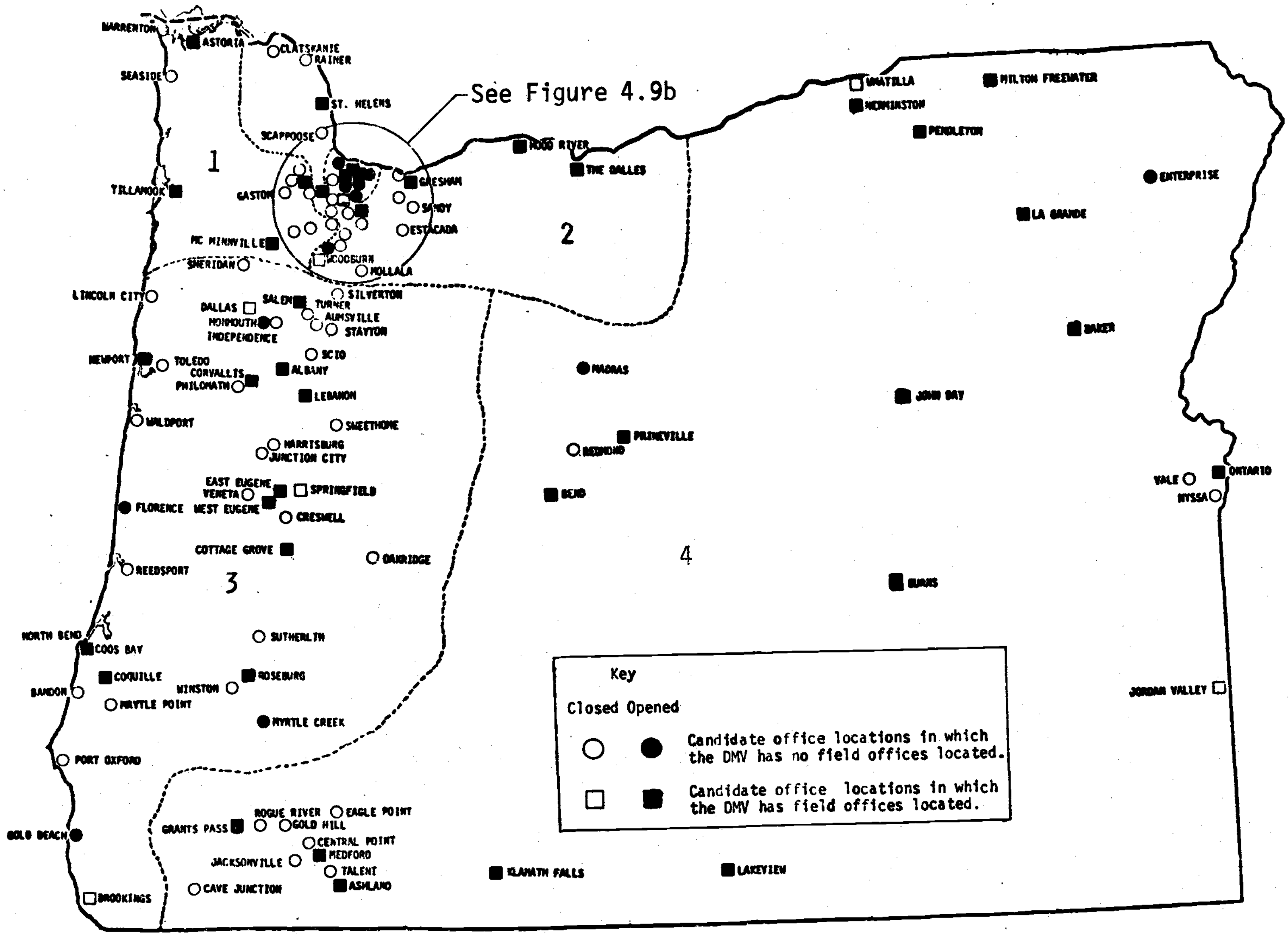


Figure 5.9a. Office locations from runs with an opening cost of $40,240 and a travel cost of $.14 per mile.

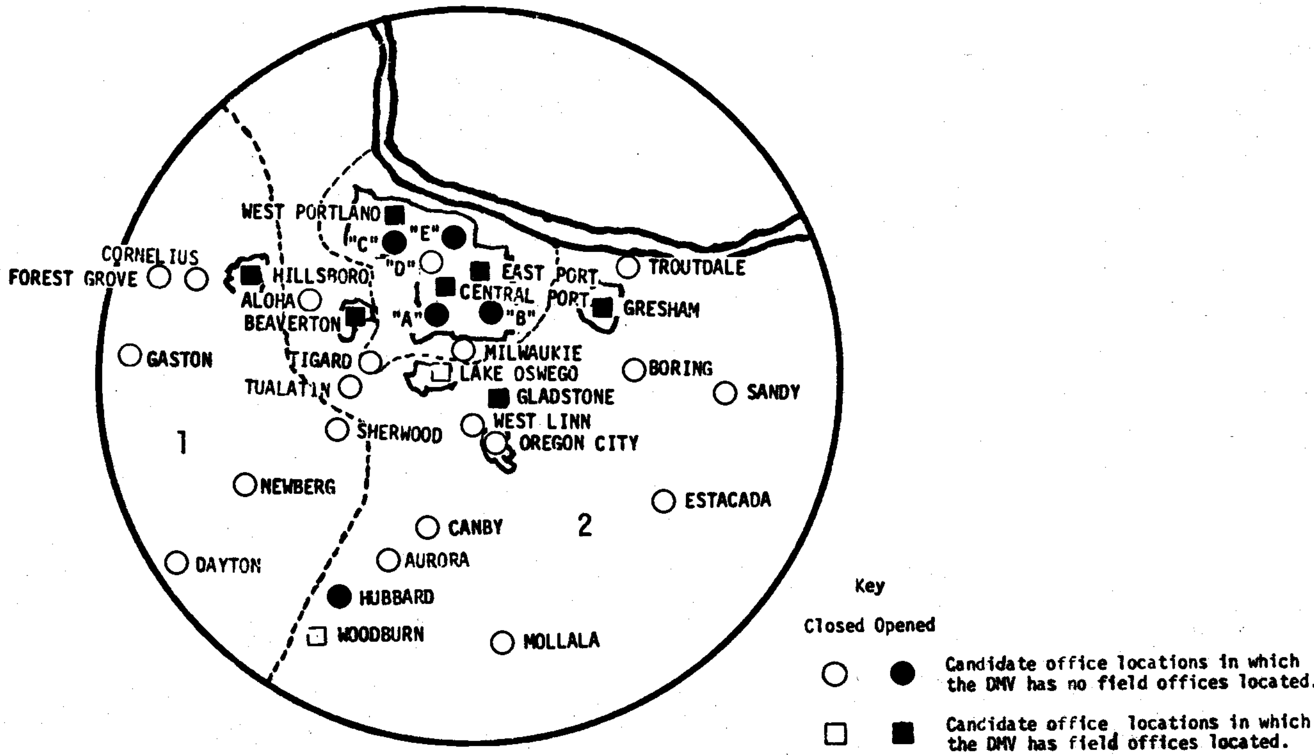


Figure 5.9b. Enlargement of Portland and the surrounding area.

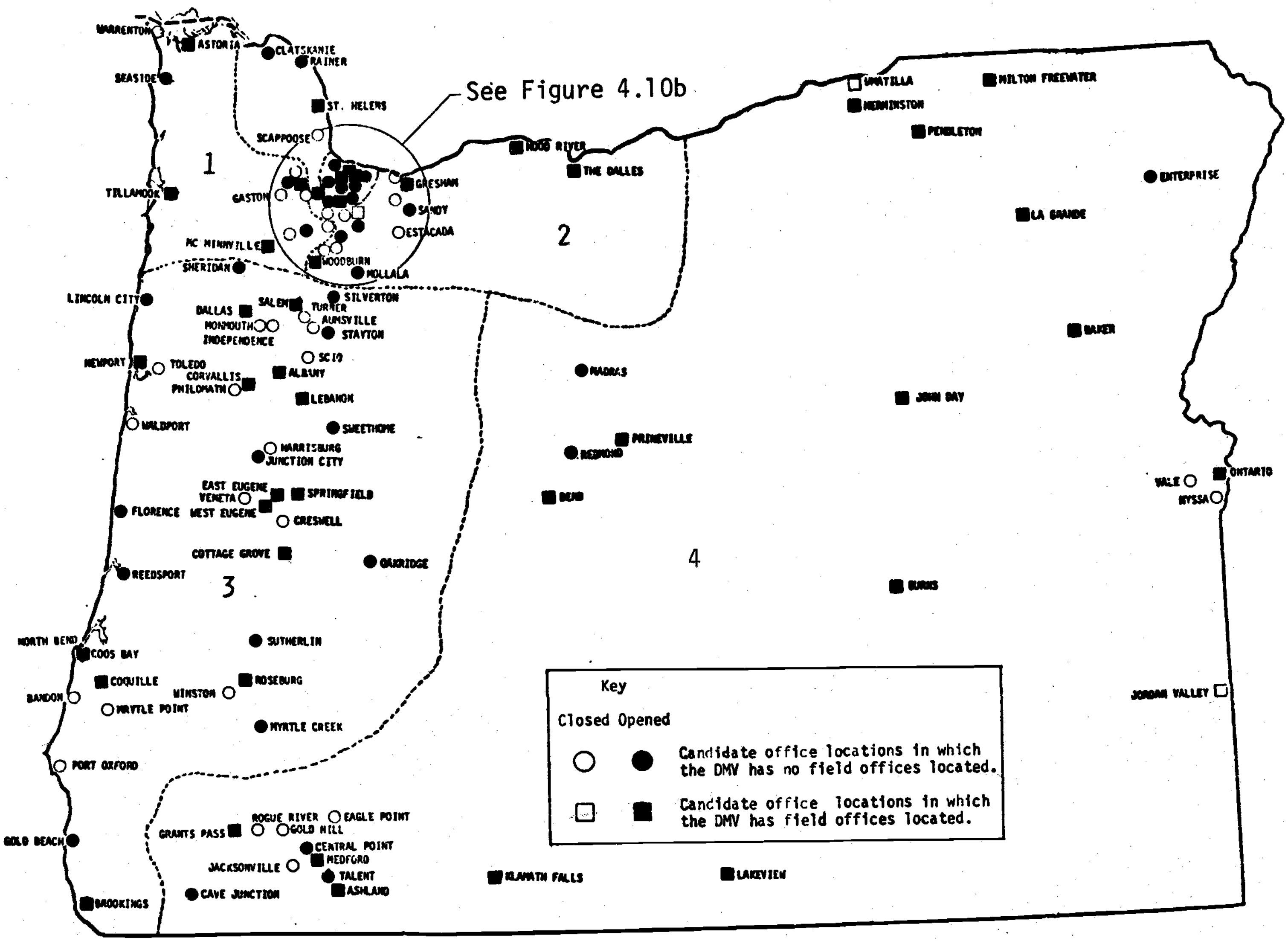


Figure 4.10a. Office locations from runs with an opening cost of $20,240 and a travel cost of $.18 per mile.

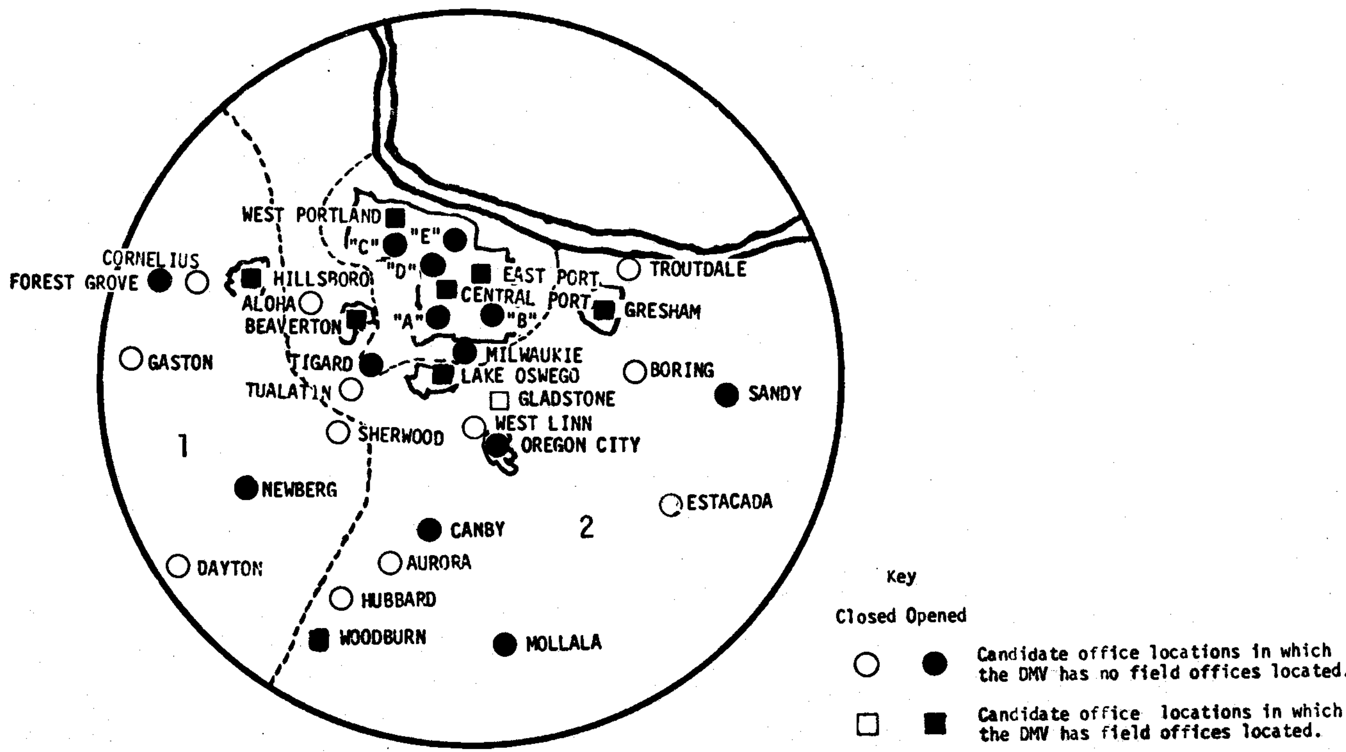


Figure 4.10b. Enlargement of Portland and the surrounding area.

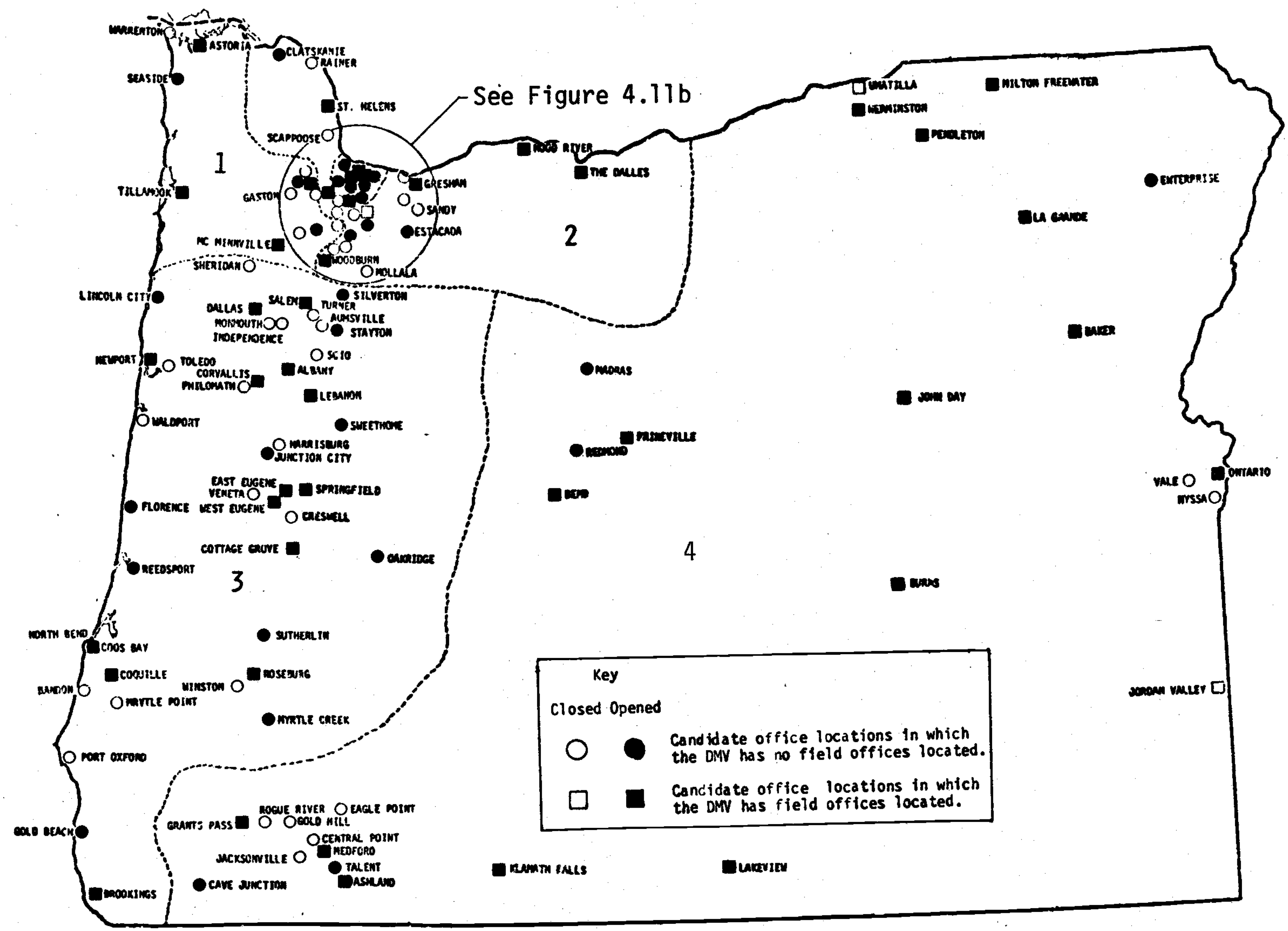


Figure 4.11a. Office locations from runs with an opening cost of $30,240 and a travel cost of $.18 per mile.

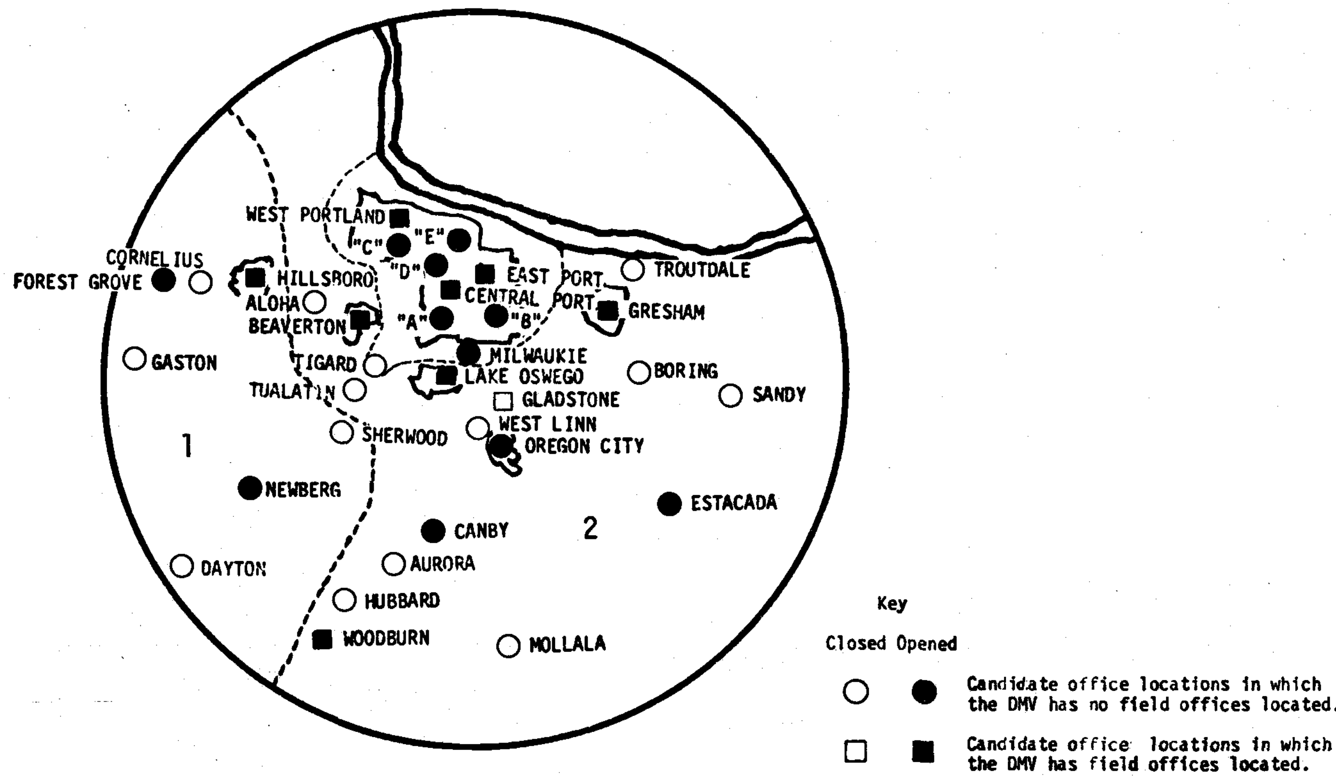


Figure 4.11b. Enlargement of Portland and the surrounding area.

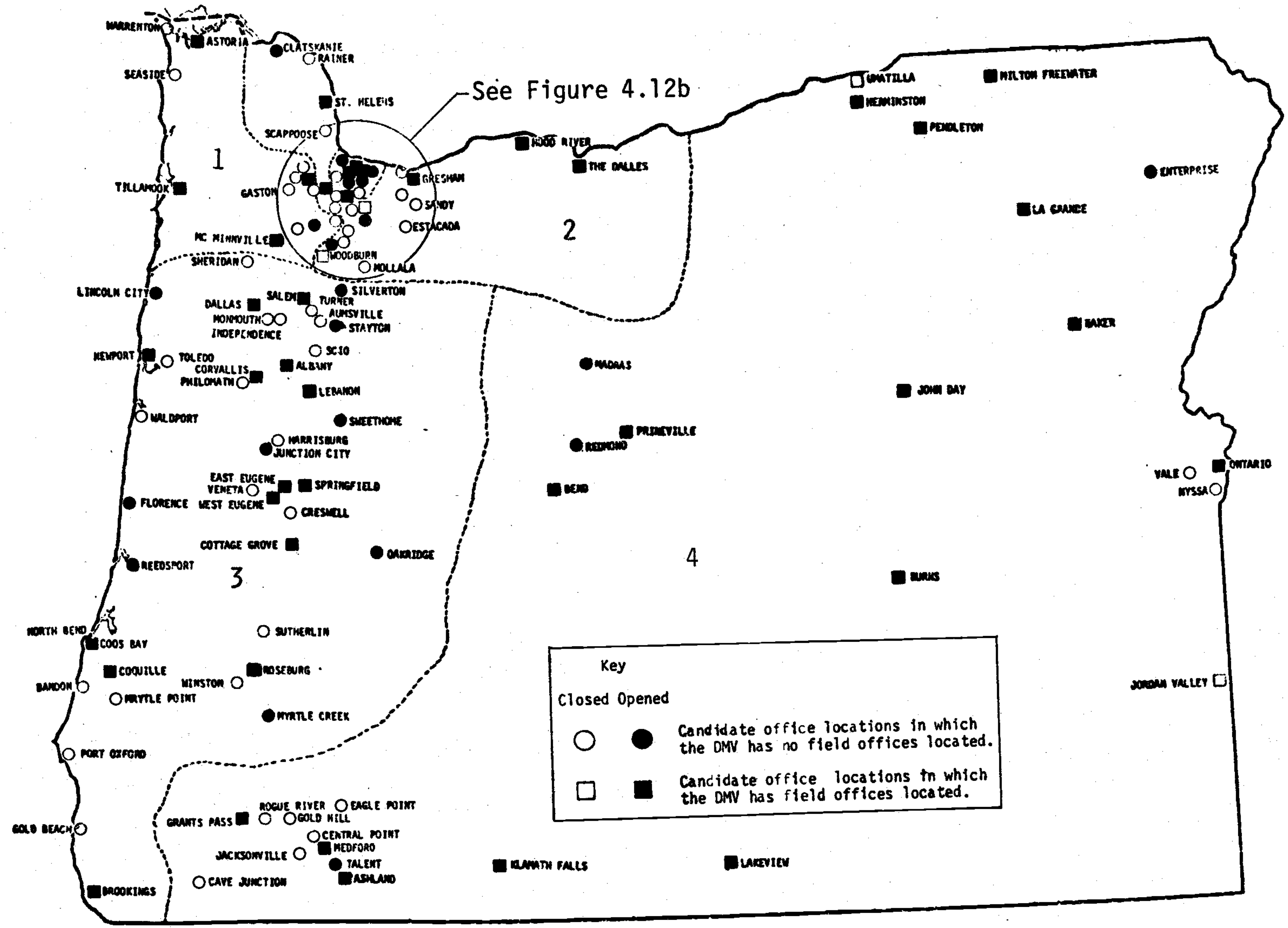


Figure 4.12a. Office locations from runs with an opening cost of $40,240 and a travel cost of $.18 per mile.

open. There are three offices open now and four more could be opened according to the results at $0.18 per mile.

4. In area 3 at $0.18 per mile, it looks as though offices should be placed in Myrtle Creek, Reedsport, Florence, Sweet Home, Lincoln City, Stayton, and Silverton. These offices should be considered for opening in the future. Silverton may have been opened because it was located on the border. If Areas 2 and 3 were combined, it might have been cheaper to close it and have the people go to a town in Area 3. Stayton is open and it has a smaller population than Silverton, so this hypothesis may not be true.

The demand for services can be determined from the solution. By knowing the demand, the approximate staffing requirement can be evaluated. A time study was done by the DMV; it was found that each employee can handle about 7,879 transactions per year. Translating this value into Drivers of Record, it becomes 7,154.[21/] Table IV shows the present staffing and the results of the runs at a travel cost of $0.10 per mile and an opening cost of $40,240 and a travel cost of $0.18 per mile and an opening cost of $30,240. The run at a travel cost of $0.10 per mile and an opening cost of $40,240 is shown because the results are close to

---

21/ Refer to Appendix C for the calculations.

TABLE VI. STAFFING REQUIREMENTS IN FTE'S

	Column			
	1	2	3	4
Office Location	Actual Requirements for 1972	Run at Costs of $.10, $40,240	Difference 1 minus 2	Run at Costs of $.18, $30,240
Beaverton	10	5.47	-4.53	5.47
Canby				2.64
Clatskanie				1.32
Estacada				1.79
Gresham	9	(X)*	-4.29	3.91
Troutdale		4.81		
Hood River	2		-2.00	1.56
Lake Oswego	3	3.00	0	3.00
Oregon City	(X)	5.96	-4.04	4.76
Gladstone	10			
St. Helens	3	3.16	+.16	3.00
The Dalles	4	3.76	-.24	2.32
Woodburn	2	(X)	+1.92	1.49
Hubbard		3.92		
Astoria	3	2.78	-.22	1.96
Forest Grove				1.88
Gaston		4.92		
Hillsboro	5	(X)	-.08	3.10

* The run at $.10 per mile traveled and $40,240 per opened office was not optimal. The demand centers on which a parentheses around an X appears would probably be opened in the optimal solution. The FTE's shown by the arrow would be needed to man the office.

TABLE VI. (cont.)

Office Location	Column 1	2	3	4
McMinnville	4	3.49	.51	2.40
Newberg				1.90
Seaside				.99
Tillamook	2	1.83	-.17	1.63
Central Portland	8	23.47		6.95
East Portland	32	14.16		6.95
West Portland			12.92*	6.95
Milwaukee				4.44
A				8.96
B		11.39		6.95
C				6.95
D				6.95
E		13.9		6.95
Salem	16	14.14	-1.96	11.64
Albany	5	4.21	-.81	4.21
Corvallis	4	4.95	+.95	4.73
Dallas	2	2.55	+.55	2.53
Stayton				2.09
Lebanon	3	4.25	1.25	2.14
Newport	4	2.48	-1.52	1.63
Lincoln City				.89

* This is the total difference for the city of Portland (Central, East, West, A, B, C, D, and E).

TABLE VI. (cont.)

Office Location	Column 1	Column 2	Column 3	Column 4
Silverton				1.26
Sweet Home				1.30
Eugene	13	19.49	-.51*	12.78
Springfield	7			4.93
Brookings	1	1.17	+.17	.71
Coos Bay				3.90
North Bend	7	4.63	-2.47	
Bandon		2.16		
Coquille	1	(X)	+1.16	1.92
Cottage Grove	2	3.13	+1.13	2.10
Florence		1.09	+1.09	1.00
Gold Beach				.69
Junction City				2.22
Myrtle Creek				1.78
Oakridge				.52
Reedsport				.81
Roseburg	7	(X)	-.31	3.88
Wilson		6.78		
Sutherlin				1.35
Medford	10	9.88	-.12	6.84
Ashland	3		-3.00	1.57

* Spring field is combine with Eugene for comparison.

TABLE VI. (cont.)

Office Location	Column 1	2	3	4
Cave Junction				.54
Grants Pass	5	5.30	+.30	4.74
Talent				1.48
Klamath Falls	7	4.91	-2.09	4.90
Lakeview	1	.71	-.29	.71
Bend	5	3.92	-1.08	2.92
Burns	2	.78	-1.22	.78
Madras				1.02
Prineville	1	2.36	1.36	1.19
Redmond				1.14
Pendleton	5	2.10	-2.90	2.10
Hermiston	2	2.18	.18	2.18
John Day	2	.82	-1.18	.82
La Grand	3	2.78	-.22	2.08
Milton Freewater	2	1.27	-.73	1.27
Baker	3	1.60	-1.40	1.60
Ontario	6	2.40	-3.60	2.40
Enterprise				.70

the present staffing. The other results are shown because it is recommended that this combination be considered in the future. An opening cost of $20,240 estimated from the 1972 budgetary summary seems low, but an opening cost of $30,240 along with a travel cost of $0.18 appears to be more representative of the actual costs.

Looking at Table VI more closely, the staffing shown in column 2[22/] appears to be fairly representative of the actual DMV requirements. This fact supports the validity of the model. The largest discrepancies in the actual and the simulated requirements exist in Portland and the surrounding areas. For example, Beaverton, Gresham and Gladstone are short four to five FTE's as is shown in column 3. But the city of Portland has an excess of 12 to 13 FTE's. The error is due to the assignment of drivers to the demand centers. Where the demand centers and offices are so close, only experience can tell what the actual needs of an office are. However, the results do help determine the number and the locations of offices which should be located in the area. The discrepancies in the other areas are partly caused by the extra FTE's included in the actual requirements to handle the Public Utility Commission and Highway transaction business. They were not excluded because only partial FTE's could be eliminated from all of the actual requirements except one. That

---

22/ The results given in column 2 are not optimal, therefore, some changes are made in a few of the locations to move the solution closer to the optimum. The changes are justified by experience from other runs. For example, the office opened in Troutdale, column 2, is shifted to Gresham.

one is Ontario where about 1.5 FTE's can be eliminated.

The optimal results given in column 4 show the office locations recommended for consideration. The same number of FTE's in column 2 can handle the offices in column 4. In actuality, more FTE's are required to handle the offices in column 4 because:

1. Of an increase in supervisory staff.
2. Of the requirement to have two men in an office. There are more offices with less than 2 men in column 4.
3. Of the difficulty to employ persons on partial FTE's. In the cases where a partial FTE is required, it would have to be increased to a full FTE.

V. CONCLUSION & RECOMMENDATIONS FOR FURTHER STUDY

The first objective of this study was to find and present a method for solving the field office location problem. After investigating several algorithms, a branch and bound algorithm proposed by Khumawala was picked. His algorithm was chosen because it gives optimal solutions, operates efficiently on a computer, and the computer code was available.

The next objective was to derive a solution. This process involved the collection of data and the actual running of the program. It was found that the algorithm ran quite efficiently, but it has storage demands which are limiting. The need for storage was minimized by dividing the problem into smaller areas. During the study a modified procedure was found, which determines a solution but does not ascertain optimality. It uses much less storage than the original branch and bound procedure and gives near optimal solutions. This procedure could be used along with one like Spielburg's (Nov. 1969) which can make use of previous solutions to assist in the determination of an optimal solution. If this were done, not as much storage capacity would be needed as was needed for the algorithm used in this study.

The final objective was to determine the feasibility of the results. The results are reasonable because:

1. In one case, the same offices which are present open were opened by the algorithm with only a couple of exceptions.

2. In other cases more offices were opened, but the locations agree with common sense.
3. The staffing requirements determined by the algorithm closely represent the actual staffing.

Besides being reasonable, the solutions are useful. The DMV is using the results to help them determine where new offices should be located. The results obtained from the runs at a travel cost of $0.18 per mile and an opening cost of $30,240 are recommended to be considered in the future as possible office locations. The staffing requirements needed to handle these offices will have to be increased to fit the actual needs.

In conclusion, the objectives of the study have been met. The recommendations for further study are:

1. To do a more detailed study in the Portland area to get a better idea of where the offices should be located within the city.
2. Also, to investigate the effects of the present location of offices by assigning a zero opening cost for the present offices.
3. To evaluate a concaved opening cost function to see what effects a large opening cost has on small offices and a small opening cost has on large offices. Before this can be done, a study would have to be made to determine the costs.
4. To study changes in demand. The DMV has made

projections evaluating the growth in demand for their services across the state. Using these projections, the future need for offices can be determined. In this way the DMV can begin preparing for changes whether there are increases or decreases in demand.

5. To make further tests on the modified branch and bound procedure, using more types of problems, to investigate its accuracy.
6. To do a larger sensitivity analysis to determine more accurately the effects shown in Figure 4.1 and 4.3. One could find out the length of the steps and the true shape of the curves. With costs increasing as they are today, the DMV should make use of these results. It should continually be reevaluating the costs associated with the operation. in order to stay abreast with the rising costs. Even during the period of time in which this paper was written, the rise in costs has made the costs used obsolete.
7. To investigate the difficulty (number of nodes used) of determining an optimal solution beyond the end points discussed in Chapter IV to get a better understanding of the algorithms behavior.

BIBLIOGRAPHY

Abernathy, William J. and John C. Hershey. 1972. A spatial-allocation model for regional health-services planning. Operations Research 20:629-642.

Center for Population Research and Census. 1972. Population estimates of counties and incorporated cities of Oregon. Portland State University.

Cooper, Leon. 1968. An extension of the generalized Weber problem. Journal of Regional Science 8:181-197.

Cooper, Leon. 1972. The transportation-location problem. Operations Research 20:94-108.

Cox, D. R. 1958. Planning of experiments. New York, John Wiley & Sons, Inc. 308 p.

Davis, P. S. and T. L. Ray. 1970. A branch-bound algorithm for the capacitated facilities location problem. Naval Research Logistics Quarterly 16:331-344.

Efroymson, M. A. and T. L. Ray. 1966. A branch and bound algorithm for plant location. Operations Research 14:361-368.

El-Shaieb. 1973. A new algorithm for locating sources among destinations. Management Science 20:221-231.

Feldman, E., F. A. Lehrer, and T. L. Ray. 1966. Warehouse location under continuous economies of scale. Management Science 12:670-684.

Hooke, Robert and T. A. Jeeves. 1961. Direct search solution of numerical and statistical problems. Journal for the Association for Computing Machinery 8:212-229.

Khumawala, Basheer M. 1970. An efficient branch and bound algorithm for warehouse location. Ph.D. dissertation, Krannert Graduate School of Industrial Administration, Purdue University 141 numb. leaves.

Khumawala, Basheer M. 1972. An efficient branch and bound algorithm for locating warehouses. Management Science 9:643-666.

Kuehn, Alfred A. and Michael J. Hamburger. 1963. A heuristic program for locating warehouses. Management Science 9:643-666.

Manne, A. A. 1964. Plant location under economies of scale-decentralization and computations. Management Science 11:213-235.

Mitten, L. G. 1970. Branch and bound methods: general formulation and properties. Operations Research 18:24-34.

Revelle, Charles S. and Ralph W. Swain. 1970. Central facilities location. Geographical Analysis 2:30-42.

Revelle, Charles S., David Marks, and John C. Liebman. 1970. An analysis of public and private sector location models. Management Science 16:692-707.

Spielburg, Kurt. Jan.-Feb. 1969. Algorithms for simple plant location models with some side conditions. Operations Research 17:85-111.

Spielburg, Kurt. Nov. 1969. Plant location with generalized search. Management Science 16:165-178.

Taha, Hamdy A. 1971. Operations Research an Introduction. New York, The Macmillan Company. 703 p.

Teitz, Michel B. and P. Bart. 1968. Heuristic methods for estimating generalized vertex median of a weighted graph. Operations Research 16:955-961.

APPENDICES

APPENDIX A

An Example

Suppose that the points located on the figure below are cities, and the circled points are candidate offices.

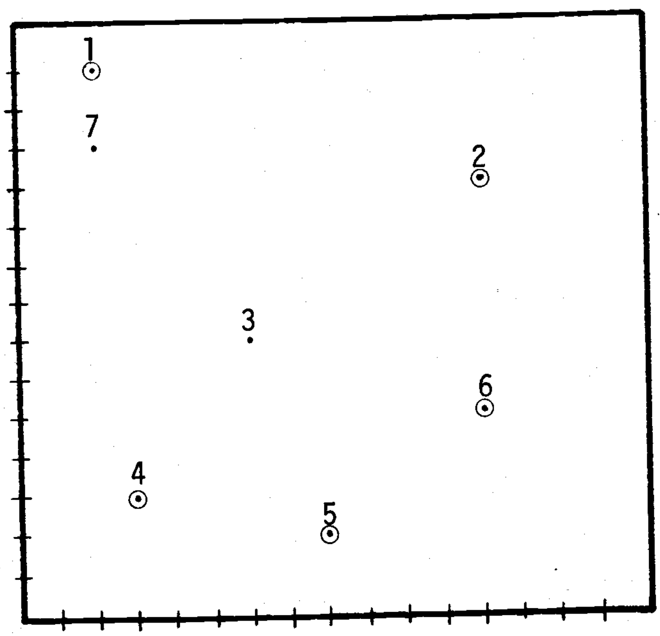


Figure A.1 Map of cities in the example

The entries in the cost matrix (TABLE II) are formed by using the following assumptions.

1. The number of miles between each point is given by:

$$\text{Miles}_{ij} = \sqrt{(Z_{1i} - X_j)^2 + (Z_{2i} - Y_j)^2} \cdot (\text{Scale})$$

where Scale = 1.875 miles/unit

(Z_{1i}, Z_{2i}) is the candidate office location.

(X_j, Y_j) is the city location.

2.

City	Location (X, Y)	Candidate Office Location (Z_1, Z_2)	Number of Drivers
1	(2, 14)	(2, 14)	200
2	(12,11)	(12,11)	100
3	(6, 7)		300
4	(3, 3)	(3, 3)	50
5	(8, 2)	(8, 2)	100
6	(12,5)	(12,5)	200
7	(2,12)		100

3. The cost matrix entry C_{ij} is given by:

$$C_{ij} = D_j \cdot [\text{Miles}_{ij} \cdot \text{Rate} + \text{Coop}]$$

where

$D_j = 1.1 \cdot$ (Number of Drivers) - This represents the number of trips to an office -- the demand.

Rate = \$.06 $\cdot$ 2 - This is the cost of travel per mile round trip.

Coop = \$1.28 - this is the cost of operating the field office per trip.

4. If a person must travel over 20 miles one way to set to a field office then a very large cost (L) will be assigned that route.

5. The cost of opening an office i is $500.00.

APPENDIX B

Nodes used for the Branch & Bound Procedures

Regular Branch & Bound Procedure

Travel Cost	Opening Cost	Area 1	2	3	4
$.10	$20,240	6	122	31	4
	$30,240	(151)*	(151)	(61)	5
	$40,240	(151)	(151)	(61)	5
$.14	$20,240	4	99	24	3
	$30,240	10	105	31	5
	$40,240	(151)	18	(61)	4
$.18	$20,240	4	16	11	3
	$30,240	6	131	(61)	3
	$40,240	14	103	32	5

*In these cases the procedure reached the storage limit.

Modified Branch and Bound

Travel Cost	Opening Cost	Area 1	2	3	4
$.10	$20,240	5	7	7	4
	$30,240	28	8	10	4
	$40,240	12	6	12	4
$.24	$20,240	4	8	7	3
	$30,240	6	7	7	4
	$40,240	6	5	10	3
$.18	$20,240	4	6	6	3
	$30,240	5	8	9	3
	$40,240	7	7	7	4

APPENDIX C

Additional estimations and assumptions used for determining field office locations.[1/]

1. Estimation of the demand for service by each driver. It is assumed that each driver of record (drivers with valid and expired licenses) goes to a field office to make his transactions. In reality some drivers make more than one trip and some make no trips (they handle their transactions by mail).

	1,938,245	- Reg./Dr. Lic. Transactions	
Less	213,281	- Dealer Title Action	- The public is not involved in these transactions.
	1,724,964	- Transactions	

$$\frac{1{,}724{,}964}{1{,}565{,}053} \quad \frac{\text{Transactions}}{\text{Drivers of Record}} = \underline{\underline{1.10}} \text{ Transactions/Dr. of Rec.}$$

This value should be close to the average number of trips.

2. Determination of the FTE's required to handle 1972 Registration and Drivers License business. The time needed to handle Public Utility Commission and Highway transaction business is eliminated from the total time. Supervisory time will be assumed to remain the same, but the fatigue and vacation time will be adjusted.

[1/] Data were obtained from a Motor Vehicles Division Report. Field services field offices' predicted staffing requirements for the 1973-75 biennium. Department of Transportation, Salem, Oregon. August 28, 1973.

Report Data			Data for Study	
Time Usage Breakdown	Hours		Hours	
Reg./Dr. Business	83,495		83,495	
Examinations	63,311		63,311	
P.U.C.	14,357			
Highway	2,293			
Supervisory	146,048	309,504	146,048	292,854
Fatigue & Vacation		99,438		
Total		409,942		

Fatigue & Vacation time represents 32.13% $(\frac{99,438}{309,504})$ of sum of the other time categories. Using this percentage the fatigue and vacation time can be calculated for the data used in the study.

The sum of the other time categories is 292,854 hours. Fatigue and vacation time = 32.13% x 292,854 = 94,094 hours. Total hours required to handle the Registration and Drivers License business is 386,948 hours.

From the Field Services field report it was determined that the DMV had a 87.74% efficiency rate. Using this and the fact that each employee has 2,016 hours available per year the number of employees needed to handle the Reg./Dr. Business can be estimated. It is:

$$\frac{386,948 \text{ hours}}{1,768.83 \text{ hours/FTE}} = \underline{\underline{218.75 \text{ FTE's}}}$$

3. Estimation of the operating cost for the DMV per Driver of Record. From the 1971-1973 budget operating expenses work out to be about \$10,120 per employee.[2/] If 218.75 FTE's are needed then the operating costs should be about:

218.75 FTE's x \$10,120 per FTE = \$2,213,750 Total Operating Cost

$$\frac{\$2{,}213{,}750}{1{,}565{,}053} = \underline{\$1.41 \text{ per Dr./Rec.}}$$ - This value is used to calculate the operating cost in the program.

4. Estimation of the number of transactions and drivers of record each employee can handle.

$$\frac{1{,}724{,}964 \text{ Transactions}}{218.75 \text{ FTE's}} = 7{,}869 \text{ Transactions/FTE}$$

$$\frac{7{,}869 \text{ Transactions/FTE}}{1.1 \text{ Transactions/Driver of Record}} = 7{,}154 \text{ Drivers of Record/FTE}$$

2/ Budget information obtained from a Motor Vehicles Division Report. DMV field office staffing and budgetary summary. Department of Transportation, Salem, Oregon. November 1, 1973.

APPENDIX D

Demand Center Location and Driver Population and
Candidate Field Office Locations

100-299 ZIP Group #1

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)[2]
1. Tigard (1)[1]	46	132	14,419
2. Amity & Perrydale	35	120	1,516
3. Arch Cape	16	147	114
4. Astoria (2)	20	160	11,535
5. Banks	38	149	1,396
6. Bay City	17	136	1,081
7. Beaver	19	128	397
8. Buxton	36	142	341
9. Cannon Beach	17	150	497
10. Carlton	35	127	1,542
11. Cloverdale	17	124	975
12. Cornelius (3)	39	135	3,029
13. Dayton (4)	38	124	2,258
14. Dundee	40	127	857
15. Forest Grove (5)	37	125	8,691
16. Gales Creek	35	138	274
17. Garibaldi	17	138	784
18. Cherry Grove	34	133	114
19. Gaston (6)	36	132	2,259
20. Glenwood	34	140	130
21. Hammond	18	160	336
22. Hebo	18	126	334

[1] Number in parentheses represent candidate office locations.

[2] Motor Vehicles Division Report. Oregon's driving population...1972. Department of Transportation, Salem, Oregon. Dec. 1973. pp. 72-84.

100-299 ZIP Group #1 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
23. Hillsboro (7)	41	135	17,771
24. Manning	36	141	120
25. Lafayette	37	127	639
26. McMinnville (8)	35	124	9,717
27. Manzanita	16	144	279
28. Nehalem	17	143	915
29. Newberg (9)	41	127	7,853
30. North Plains	41	138	445
31. Oceanside	16	134	145
32. Pacific City	15	125	166
33. Rockaway & Manhattan Beach	16	139	949
34. St. Paul	41	124	611
35. Gearhart	17	154	633
36. Seaside (10)	17	153	4,520
37. Sherwood (11)	45	129	4,307
38. Tillamook (12)	18	133	6,481
39. Netarts	16	133	185
40. Timber	33	143	92
41. Tolovana Park	17	149	135
42. Warrenton (13)	18	160	2,140
43. Wheeler	18	142	240
44. Yamhill	35	129	1,501
45. Newosin	14	121	164
46. Central Portland (14)	49	134	49,719
47. East Portland (15)	48	137	49,719
48. West Portland (16)	51	136	49,719
49. Milwaukie (17)	50	132	31,796
50. A Portland (18)	48	133	49,719
51. B Portland (19)	51	133	49,719
52. C Portland (20)	47	136	49,719
53. D Portland (21)	49	135	49,719
54. E Portland (22)	50	137	49,719

00-99 ZIP Group #2

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
1. Antelope & Shaniko (1)	99	111	145
2. Aurora	46	124	2,659
3. Beaver Creek	53	126	1,491
4. Beaverton (2)	46	134	31,478
5. Aloha (3)	44	134	7,692
6. Bonneville	68	139	257
7. Boring (4)	57	132	5,780
8. Bridal Veil	62	136	110
9. Brightwood	66	129	380
10. Canby (5)	48	125	6,386
11. Cascade Locks	70	140	583
12. Clackamas (6)	52	131	4,767
13. Birkenfield & Mist	34	153	207
14. Clatskanie (7)	36	157	3,276
15. Westport & Brownsmead	31	158	354
16. Colton	55	122	1,127
17. Columbia City	45	149	387
18. Donald	44	124	179
19. Dufur	89	131	675
20. Eagle Creek	57	129	1,440
21. Estacada (8)	57	126	4,588
22. Fairview	54	136	702
23. Gervais	42	120	1,491
24. Gladstone & Jennings Lodge (9)	51	130	5,221
25. Gov't Camp	72	127	160
26. Grass Valley	98	128	301
27. Gresham (10)	55	134	15,754
28. Hood River (11)	79	141	8,105
29. Hubbard (12)	46	123	2,077
30. Kent	100	122	98
31. Lake Grove	48	131	1,029
32. Lake Oswego & Oak Grove (13)	49	132	18,301

00-99 ZIP Group #2 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
33. Maupin	90	121	815
34. Molalla (14)	51	121	4,783
35. Moro	99	132	445
36. Mosier	82	140	456
37. Mt. Hood	78	135	113
38. Mulino	51	124	1,470
39. Odell	79	138	300
40. Oregon City (15)	51	129	15,528
41. Parkdale	77	134	1,356
42. Rainier & Goble (16)	43	156	3,364
43. Rhododendron	79	127	230
44. Rufus	99	140	312
45. St. Helens (17)	46	147	6,200
46. Warren	44	146	1,545
47. Deer Island	45	150	793
48. Sandy (18)	59	130	4,523
49. Scappoose (19)	43	144	4,186
50. The Dalles (20)	88	137	12,367
51. Troutdale (21)	56	135	5,601
52. Tualatin (22)	46	130	2,121
53. Tygh Valley	88	124	505
54. Wamic & Friend	86	123	104
55. Vernonia	36	148	1,765
56. Wasco	100	136	593
57. Wemme	67	128	371
58. West Linn (23)	50	129	7,019
59. Wilsonville	46	127	1,489
60. Woodburn & Monitor (24)	43	121	8,960
61. Zigzag	68	128	195

300-499 ZIP Group #3

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
1. Yoncalla	10	92	894
2. Yachats	30	65	1,070
3. Alsea	23	93	841
4. Brooks	41	117	533
5. Salem (1)	39	114	82,763
6. Agate Beach	11	105	132
7. Albany (2)	36	102	23,965
8. Aumsville (3)	42	110	2,042
9. Blodgett	26	101	346
10. Brownsville	39	94	1,475
11. Burntwood	23	102	148
12. Cascadia	53	93	114
13. Corvallis (4)	32	100	28,412
14. Crabtree	42	103	156
15. Crawfordsville	42	92	267
16. Dallas (5)	31	113	7,581
17. Depoe Bay	12	110	674
18. Detroit	62	105	245
19. Eddyville	19	103	266
20. Falls City	29	111	553
21. Foster	48	94	523
22. Gates	54	106	336
23. Grande Ronde	24	110	720
24. Halsey	36	93	964
25. Idanha	63	104	391
26. Independence (6)	35	100	3,180
27. Jefferson	30	106	1,928
28. Kings Valley	28	105	58
29. Lacomb	45	100	484
30. Stayton (7)	45	108	3,254
31. Lebanon (8)	41	99	13,172
32. Lyons	49	107	1,330

300-499 ZIP Group #3 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
33. Marion	38	107	214
34. Mill City	53	106	1,160
35. Monmouth (9)	33	110	3,832
36. Mt. Angel & Marquam	45	118	1,910
37. Neotsu	14	117	221
38. Newport (10)	11	104	5,517
39. Southbeach	11	103	544
40. Lincoln City & Kernville (11)	13	115	3,324
41. Otis	15	117	1,007
42. Otter Rock	11	108	134
43. Philomath & Nashville (12)	29	99	3,383
44. Rickreal	34	113	495
45. Rose Lodge	18	118	99
46. Scio (13)	43	105	2,695
47. Logsden	19	107	139
48. Scotts Mills	48	117	559
49. Seal Rock	10	98	410
50. Shedd	36	96	683
51. Sheridan (14)	30	120	2,953
52. Siletz	16	106	710
53. Silverton (15)	45	116	6,531
54. Mehama	49	108	303
55. Sublimity	45	110	935
56. Sweet Home (16)	46	94	6,791
57. Bleneden Beach	12	112	414
58. Tangent	36	99	816
59. Waldport	15	95	223
60. Toledo (17)	14	103	3,467
61. Turner (18)	41	110	2,284
62. Valsetz	23	111	272
63. Willamina	27	119	1,716
64. Camas Valley	19	44	601

300-499 ZIP Group #3 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
65. Port Orford	-6	35	1,446
66. Eugene (19)	36	81	91,444
67. Coburg	37	84	541
68. Pleasent Hill	40	77	1,586
69. Leaburg	47	83	632
70. McKenzie Bridge	61	85	286
71. Finn Rock	55	84	89
72. Goshen	38	78	133
73. Jasper	41	79	144
74. Agness	8	27	74
75. Allegany	10	60	74
76. Alvadore	51	83	133
77. Azalea	30	34	386
78. Bandon (20)	-1	48	3,226
79. Blachly	25	87	276
80. Blue River	56	84	531
81. Broadbent	7	44	216
82. Brookings & Harbor (21)	1	9	5,111
83. Canyonville	29	40	1,291
84. Chesire	31	87	526
85. Charleston	2	56	580
86. Coos Bay (22)	5	57	16,890
87. Eastside	6	57	1,030
88. Coquille (23)	5	50	5,352
89. Cottage Grove & Saginaw (24)	36	72	8,842
90. Cresent Lake	65	60	80
91. Creswell & Disston (25)	37	76	3,457
92. Culp Creek	43	68	259
93. Curtin	32	69	103
94. Days Creek	23	41	488
95. Deadwood	17	82	176
96. Dexter	43	76	1,369
97. Dillard	26	46	440

300-499 ZIP Group #3 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
98. Dorena	42	69	343
99. Drain	29	67	1,695
100. Elkton	23	66	563
101. Elmira	29	82	1,234
102. Fall Creek	43	78	626
103. Florence (26)	9	80	4,354
104. Gardiner	9	70	399
105. Glendale	25	33	1,431
106. Glide	35	52	1,443
107. Postal River	-1	17	82
108. Gold Beach (27)	-2	22	2,927
109. Greenleaf	17	82	99
110. Harrisburg (28)	34	89	1,956
111. Idleyld Park	37	55	562
112. Junction City (29)	33	86	6,200
113. Horton	26	89	119
114. Lakeside	7	65	973
115. Lanlois	-2	41	454
116. Lorane	32	73	295
117. Lowell	44	76	663
118. Mapleton & Tiernan	15	81	960
119. Marcola	42	86	731
120. Milo	25	40	155
121. Monroe & Alpine	31	91	1,327
122. Mrytle Creek (30)	29	43	5,042
123. Mrytle Point & Norway (31)	6	46	3,610
124. North Bend (32)	5	59	9,329
125. Noti	26	81	419
126. Oakland	30	58	1,873
127. Oakridge	52	69	3,096
128. Ophir	1	38	131
129. Powers, Gaylord, & Remote	8	39	775
130. Reedsport	8	69	3,838

300-499 ZIP Group #3 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
131. Winchester Bay	7	69	396
132. Riddle	27	41	1,886
133. Roseburg (35)	28	51	23,953
134. Scottsburg	16	68	220
135. Sizes	-5	37	237
136. Springfield (36)	38	81	28,811
137. Sunny Valley	27	29	337
138. Sutherlin (37)	29	57	3,655
139. Swisshome	17	82	391
140. Tenmile	22	46	311
141. Tiller	38	39	276
142. Umpqua	25	56	324
143. Veneta (38)	29	81	2,665
144. Vida	50	83	562
145. Walton	23	81	174
146. Wedderburn	-1	24	147
147. Westfir	52	76	569
148. Westlake	9	87	279
149. Wilbur	29	54	195
150. Winchester	28	53	560
151. Winston (39)	26	47	2,032
152. Wolf Creek	26	31	381

500-999 ZIP Group #4

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
1. Arock	184	40	51
2. Eagle Pt. (1)	42	23	3,856
3. Central Pt. (2)	38	19	9,786
4. Medford (3)	40	17	35,121

500-999 ZIP Group #4 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
5. White City	41	21	1,318
6. Ashland (4)	44	12	11,258
7. Butte Falls	42	25	468
8. Cave Junction (5)	18	12	2,228
9. Gold Hill (6)	35	22	2,408
10. Grants Pass (7)	28	22	26,310
11. Applegate	22	14	343
12. Jacksonville (8)	37	17	2,899
13. Kerby	19	13	333
14. Merlin	24	24	706
15. O'Brien	17	8	264
16. Phoenix	41	15	1,105
17. Prospect	51	32	725
18. Rogue River (9)	32	21	2,520
19. Selma	19	16	725
20. Shady Cove	42	28	932
21. Talent (10)	42	24	2,603
22. Trail	42	30	648
23. Wilderville	23	20	339
24. Williams	29	14	623
25. Klamath Falls (11)	70	12	30,690
26. Crater Lake	60	37	57
27. Adel	121	10	118
28. Beatty	83	20	165
29. Bly	90	18	386
30. Chiloquin	68	36	1,206
31. Dairy	77	13	136
32. Fort Klamath	64	30	157
33. Keno	65	9	398
34. Lakeview (12)	109	11	3,977
35. Malin	80	5	958
36. Merrill	25	25	1,128
37. Midland	69	9	242

500-999 ZIP Group #4 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
38. New Pine Creek	110	4	130
39. Paisley	104	29	363
40. Plush	121	19	51
41. Silver Lake & Christmas Valley	90	46	444
42. Sprague River	77	21	186
43. Summer Lake	97	40	69
44. Bonanza	80	11	944
45. Culver	86	97	821
46. Crane	156	56	121
47. Fort Rock	90	54	68
48. Hines	143	62	1,167
49. Bend (13)	83	80	18,094
50. Andrews	156	31	12
51. Ashwood	99	105	73
52. Brothers	102	71	50
53. Burns (14)	144	63	3,599
54. Princeton	156	52	122
55. Diamond	154	41	80
56. Camp Sherman	75	94	107
57. Chemult	70	50	353
58. Diamond Lake	61	48	14
59. La Pine	78	66	1,063
60. Lawen	150	57	50
61. Madras (15)	88	101	4,443
62. Cresent	73	58	407
63. Gilchrist	73	59	492
64. Metolius	87	99	209
65. Mitchell	114	98	300
66. Paulina	120	82	137
67. Post	107	83	55
68. Powell Butte	92	87	455
69. Prineville (16)	96	89	7,534
70. Redmond (17)	87	88	6,134

500-999 ZIP Group #4 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
71. Riley	132	60	64
72. Sisters	77	89	853
73. Terrebonne	87	91	1,033
74. Warm Springs	85	107	840
75. Frenchglen	147	35	59
76. Pendleton & Rieth (18)	149	139	12,729
77. Adams	155	143	502
78. Alicel	170	130	185
79. Arlington & Olex	114	131	545
80. Cecil	129	137	37
81. Condon	113	123	1,015
82. Cove	175	126	640
83. Dayville	130	95	217
84. Enterprise (19)	188	131	2,137
85. Echo	139	142	729
86. Elgin	172	136	1,690
87. Fossil	112	114	533
88. Haines	172	112	602
89. Halfway	194	111	881
90. Helix	152	145	313
91. Heppner	129	128	1,523
92. Hereford	170	97	102
93. Hermiston (20)	136	145	7,532
94. Lexington	126	131	309
95. Imbler	171	132	244
96. Imnaha	200	137	126
97. Ione	123	133	512
98. Irrigon	131	147	585
99. John Day (21)	146	93	1,868
100. Joseph	190	129	1,211
101. Kimberly	128	106	120
102. Kinzua	116	114	394
103. La Grande (22)	167	127	9,795

500-999 ZIP Group #4 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
104. Island City	168	128	281
105. Long Creek & Fox	142	104	325
106. Lostine	184	134	295
107. McNary	137	148	181
108. Meacham	159	134	91
109. Mikkalo	112	132	33
110. Milton Freewater & Umapine (23)	159	149	6,992
111. Athena	157	144	878
112. Baker (24)	175	107	8,150
113. Bates	158	100	267
114. Boardman	126	145	471
115. Monument	134	108	188
116. Mt. Vernon	141	93	603
117. North Powder	172	116	472
118. Pilot Rock	148	132	1,686
119. Prairie City	152	94	919
120. Richland	192	107	515
121. Senca	145	83	299
122. Spray	124	108	245
123. Stanfield	138	143	1,118
124. Summerville	169	133	434
125. Sumpter	165	105	152
126. Telocaset & Medical Springs	177	117	55
127. Ukaih	146	119	222
128. Dale & Ritter	146	114	177
129. Umatilla	135	148	1,031
130. Union	174	123	1,538
131. Unity	166	94	242
132. Wallawa	182	136	1,213
133. Weston	159	144	749
134. Bridgeport	178	96	41
135. Canyon City	146	92	672
136. Cayuse	156	140	83

500-999 ZIP Group #4 (cont'd.)

Customer Code	Coordinates X	Coordinates Y	Drivers of Record (1972)
137. Adrian	196	69	224
138. Drewsey	162	71	138
139. Durkee	185	100	125
140. Harper	182	73	234
141. Huntington	190	92	551
142. Ironside	173	90	88
143. Jamieson	186	86	84
144. Jordan Valley (25)	198	41	431
145. Ontario (26)	198	80	8,773
146. Vale & Willow Creek (27)	191	78	2,907
147. Westfall	179	78	54
148. Juntura & Riverside	169	69	140
149. Nyssa (28)	198	75	3,751
150. Brogan	184	89	90

APPENDIX E

The Computer Code

```
FORTRAN IV G1  RELEASE 2.0            MAIN              DATE = 74193          15/03/14
0001              REAL IFC,ID,IVC,MDEL,MDELS,MINC,MEGAS,JD,LLN,MINC1,MINC2,IJD
0002              DIMENSION IFC( 24),ID(61),IVC( 24,61),MDEL(150,61),MDELS(150,24)
                 1,MEGAS(150,24),JD(150,24),MINC( 61),Z(150),Y(150,24),IJD( 24),
                 2ISOL(150,61),X( 61),YX( 61),D(161),ZX(2, 61)
                 3,OFF( 24),OFFF( 24)
                 4,KZ(150,24),K1(150,24),K2(150,24),LN(150,24)
                 5,IDEL(150,61),ILN( 24)
0003              EQUIVALENCE (MDEL(1,1),ZX(1,1))
                 1,(MINC(1),X(1)),(MDELS(1,1),YX(1)),(JD(1,1),D(1))
            C
            C
            C  DICTIONARY OF THE MAIN TERMS USED IN THE PROGRAM
            C
            C       K2 - THE SET OF OFFICES NOT OPENED OR CLOSED %FREE¤
            C       K1 - THE SET OF OFFICES THAT HAVE BEEN OPENED
            C       KZ - THE SET OF OFFICES THAT HAVE BEEN CLOSED
            C       LN - THE SET OF CUSTOMERS WHICH CAN BE SUPPLIED BY OFFICE IW
            C       IDEL - THE SET OF OFFICES THAT HAVE BEEN OPENED AS A RESULT OF
            C         THE DELTA CALCULATIONS AND THEIR RESPECTIVE CUSTOMERS
            C       IFC - FIXED OFFICE COST
            C       ID - DEMAND FOR SERVICE
            C       IVC - THE VARIABLE COSTS RESULTING FROM TRAVEL COSTS AND
            C         OPERATING COSTS
            C       MDEL - DELTA
            C       MDELS - SUM OF THE DELTAS FOR A SPECIFIC OFFICE AND NODE
            C       MEGAS - OMEGAS
            C       JD - THE SET OF OFFICES WHICH CAN SUPPLY CUSTOMER IC
            C       Z - TOTAL COST
            C       Y - EQUALS 0 OF THE OFFICE IS CLOSED AND 1 IF THE OFFICE IS
            C         OPEN
            C       SOL - THE SET OF OPEN OFFICES IN THE TERMINAL SOLUTIONS
            C       NW - THE NUMBER OF POSSIBLE OFFICE LOCATIONS
            C       NC - THE NUMBER OF CUSTOMERS
            C       UBD - UPPER BOUND
            C       LBD - LOWER BOUND
            C       NUBDN - NEW UPPER BOUND NODE
            C       NLBDN - NEW LOWER BOUND NODE
            C       MODE - NUMBER OF DISTINCT NODES INVESTIGATED
            C       ITER - INTERATIONS
            C    THESE VARIABLES MAY BE CHANGED
            C       RATE - THE COST OF TRAVEL PER MILE FOR A CUSTOMER
            C       XRATE - THE NUMBER OF TIMES THAT EACH DRIVER OF RECORD GOES
            C         TO THE FIELD OFFICE
            C       DRPEMP - THE NUMBER OF DRIVERS THAT EACH EMPLOYEE CAN HANDLE
            C       VOCOST - THE COST TO THE DMV PER DRIVER OF RECORD TO PROVIDE
            C         ITS SERVICES
            C       FWCOST - THE COST OF OPENING A FIELD OFFICE
            C
```

```
FORTRAN IV G1  RELEASE 2.0            MAIN              DATE = 74193          15/03/14
                C  SOME COMMENTS ABOUT THE INPUT DATA
                C        1. THE NUMBER OF CUSTOMERS AND THE NUMBER OF POSSIBLE OFFICE
                C           LOCATIONS MUST BE SUPPLIED.
                C        2. THE COORDINATE %X,Y¤ LOCATIONS AND THE NUMBER OF DRIVERS OF
                C           RECORD AT EACH LOCATION MUST BE PROVIDE.
                C        3. THE COORDINATES OF THE FIELD OFFICES MUST BE GIVEN.
                C
                C
                C  AN EXPLANATION OF THE ALGORITHM USED TO SOLVE THIS PROBLEM CAN BE
                C   FOUND IN AN ARTICLE BY KHUMAWALA, B. M., @AN EFFICIENT BRANCH AND
                C   BOUND ALGORITHM FOR THE WAREHOUSE LOCATION PROBLEM,@ @@MANAGEMENT
                C   SCIENCE@@, VOL. 18, NO. 12, %AUGUST 1972¤.
                C
                C
                C  THE COMPUTER PROGRAM CAN BE FOUND IN AN UNPUBLISHED PH.D.
                C   DISSERTATION.  KHUMAWALA, B. M., @AN EFFICIENT BRANCH AND BOUND
                C   ALGORITHM FOR WAREHOUSE LOCATION@, KRANNERT GRADUATE SCHOOL OF
                C   INDUSTRIAL ADMINISTRATION, PURDUE UNIVERSITY, JUNE 1970.
                C
                C
 0004           20001 FORMAT(3(3F8.0))
 0005           20002 FORMAT(5(2F8.0))
 0006           303   FORMAT(//////44X,'FIELD OFFICES OPENED FOR THE DMV',//)
 0007           5833  FORMAT(//5X,'FIRST TERMINAL SOLUTION FOUND WAS',F10.2,2X,'IT WAS F
                     1OUND AT INTERATION NUMBER',I15)
 0008           10001 FORMAT(2I10)
 0009           10003 FORMAT(///5X,'SOLUTION INFEASIBLE')
 0010           10004 FORMAT(///2X,'THE OPTIMAL SOLUTION FOUND AFTER',I7,'    ITERATIONS
                     1 IS ',F15.2,'  IT WAS FOUND AT ITERATION NUMBER',I10,/5X,'MAXIMUN
                     1NUMBER OF DISTINCT NODES USED',I10)
 0011           10005 FORMAT(//3X,A4,A4,5X,'SUPPLIES THE FOLLOWING CUSTOMERS')
 0012           10006 FORMAT(/10X,I10,10X,'AT A COST OF',5X,F10.2)
 0013           10007 FORMAT(//,35X,'THE COST TO THE PUBLIC IS',F5.2,2X,'PER MILE.',/,
                     135X,'THE COST TO THE DMV TO PROVIDE THE SERVICE IS',F5.2,2X,'PER D
                     2RIVER OF RECORD.',/,35X,'THE COST TO OPEN A FIELD OFFICE IS',F9.1)
 0014           10008 FORMAT(/9X, A4,A4,2X,'REQUIRES',F6.2,2X,'STAFF.')
 0015           10009 FORMAT(5X,A4,A4)
 0016           57791 FORMAT(//5X,'COMPUTATIONS DISCONTINUED FOR MORE STORAGE.  SOLTION
                     1GIVEN BELOW MAY NOT NECESSARILY BE OPTIMAL')
 0017                 READ(5,10001)NW,NC
 0018                 READ(5,20001) (X(I),YX(I),D(I),I=1,NC)
 0019                 READ(5,20002) (ZX(1,I),ZX(2,I),I=1,NW)
 0020                 READ(5,10009) (OFF(J),OFFF(J),J=1,NW)
 0021                 DRPEMP=7154.53
 0022                 XRATE=1.10
 0023                 VOCOST=1.41
 0024                 RATE=.06
 0025                 DO 99 IJK=1,3
```

```
              FWCOST=10240.
              RATE=RATE + .04
              DO 99 JKL=1,3
              FWCOST=FWCOST + 10000.
              IF(JKL.EQ.1.AND.IJK.EQ.1) GO TO 991
              DO 2122 IW=1,NW
              IFC(IW)=FWCOST
              DO 2122 IC=1,NC
              D(IC)=ID(IC)/XRATE
              XMILES=(IVC(IW,IC)-D(IC)*VOCOST)/(ID(IC)*YRATE*2.)
              IF(XMILES.LE.150.) GO TO 4
              IVC(IW,IC)=9.E38
              GO TO 2122
     4        IVC(IW,IC)=(ID(IC)*XMILES*RATE*2.)+D(IC)*VOCOST
     2122     CONTINUE
              GO TO 992
     991      CONTINUE
              DO 2121 IW=1,NW
              IFC(IW)=FWCOST
              DO 2121 IC=1,NC
              XMILES=((ZX(1,IW)-X(IC))**2)+((ZX(2,IW)-YX(IC))**2)
              IF(XMILES.EQ.0.) GO TO 5
              XMILES=SQRT(XMILES)*1.875
     5        CONTINUE
              IF(XMILES.LE.150.) GO TO 3
              IVC(IW,IC)=9.E38
              GO TO 2121
     3        ID(IC)=D(IC)*XRATE
              IVC(IW,IC)=(ID(IC)*XMILES*RATE*2.)+D(IC)*VOCOST
     2121     CONTINUE
     992      CONTINUE
              YRATE=RATE
              METHOD=3
              IF(METHOD.EQ.3)WRITE(6,303)
              WRITE(6 ,10007) RATE,VOCOST,FWCOST
     C
     C        INITIALIZATION
     C
              NFIRST=0
              NKTR=0
              NKTR1=0
              LLN=9.999E38
              XLBD=0.0
              UBD=LLN
              MODE=1
              NODE=1
              NUBDN=NODE
              ITER=1
```

```
0071                KODE=0
0072                DO 1000 IW=1,NW
0073                JD(NODE,IW)=0
0074                KZ(NODE,IW)=0
0075                K1(NODE,IW)=0
0076                K2(NODE,IW)=1
0077                LN(NODE,IW)=NC
0078                DO 1001 IC=1,NC
0079                JD(NODE,IW)=JD(NODE,IW)+ID(IC)
0080                IF(IW.GE.2)GO TO 1001
0081                IDEL(NODE,IC)=0
0082          1001  CONTINUE
0083                ILN(IW)=LN(NODE,IW)
0084                IJD(IW)=JD(NODE,IW)
0085          1000  CONTINUE
0086                GO TO 786
              C
              C     SETS ARE UPDATED
              C
0087             1  CONTINUE
0088                ITER=ITER+1
0089                IF(NLBDN.EQ.1)GO TO 4193
0090                IF(NKTR.EQ.1.OR.NKTRI.EQ.1) GO TO 4192
0091                IF(KODE.NE.0)GO TO 4195
0092          4193  NODE=MODE+1
0093                MODE=NODE
              C
              C     STORAGE ALLOTMENT CHECK
              C
0094                IF(MODE.GT.150)GO TO 9779
0095                GO TO 4196
0096          4195  NODE=KODE
0097                KODE=0
0098          4196  DO 5167 IC=1,NC
0099                IDEL(NODE,IC)=IDEL(NLBDN,IC)
0100                MDEL(NODE,IC)=MDEL(NLBDN,IC)
0101          5167  CONTINUE
0102                DO 92 IW=1,NW
0103                JD(NODE,IW)=JD(NLBDN,IW)
0104                KZ(NODE,IW)=KZ(NLBDN,IW)
0105                K1(NODE,IW)=K1(NLBDN,IW)
0106                K2(NODE,IW)=K2(NLBDN,IW)
0107                LN(NODE,IW)=LN(NLBDN,IW)
0108                MDELS(NODE,IW)=MDELS(NLBDN,IW)
0109                MEGAS(NODE,IW)=MEGAS(NLBDN,IW)
0110            92  CONTINUE
0111                GO TO 4194
0112          4192  NODE=NLBDN
```

```
      4194 IF(NKTR.EQ.0)GO TO 3786
           GO TO (3912,3911),NKTR
      3786 GO TO (911,912),NKTR1
      3911 NKTR=NKTR-1
           GO TO 3913
      911  NKTR1=NKTR1-1
      3913 KZ(NODE,KKW)=1
           K2(NODE,KKW)=0
           GO TO 786
      3912 NKTR=NKTR-1
           GO TO 3914
      912  NKTR1=NKTR1-1
      3914 K1(NODE,KKW)=1
           K2(NODE,KKW)=0
           GO TO 787
C
C          SIMPLIFICATION CYCLE
C
      786  CONTINUE
           DO 20 IC=1,NC
           KKK=0
           KTR=0
           DO 10 IW=1,NW
           IF(KZ(NODE,IW).EQ.1)GO TO 10
           IF(K1(NODE,IW).EQ.1.AND.IDEL(NODE,IC).EQ.IW)GO TO 20
           KTR=KTR+1
           IF(KTR.EQ.1) GO TO 11
           IF(KTR.EQ.2) GO TO 12
           IF(IVC(IW,IC).GE.MINC2) GO TO 10
           GO TO 12
      11   MINC1=IVC(IW,IC)
           MW=IW
           GO TO 10
      12   CONTINUE
           MINC1=AMIN1(MINC1,IVC(IW,IC))
           IF(MINC1.EQ.IVC(IW,IC)) GO TO 13
           MINC2=IVC(IW,IC)
           GO TO 10
      13   MINC2=IVC(MW,IC)
           MW=IW
      10   CONTINUE
           IF(KTR.EQ.0) GO TO 19
           IF(KTR.EQ.1) GO TO 14
           IDEL(NODE,IC)=MW
           MDEL(NODE,IC)=MINC2-MINC1
           GO TO 20
      14   K1(NODE,MW)=1
           K2(NODE,MW)=0
```

```
                KKK=KKK+1
                GO TO 20
      C
      C         FEASIBILITY CHECK
      C
      19        IF(NODE.NE.1) GO TO 74
                WRITE(6,10003)
                STOP
      20        CONTINUE
                KTR=KKK
                DO 25 IW=1,NW
                IF(K2(NODE,IW).EQ.0) GO TO 25
                MDELS(NODE,IW)=-IFC(IW)
                DO 30 IC=1,NC
                IF(IDEL(NODE,IC).NE.IW) GO TO 30
                MDELS(NODE,IW)=MDELS(NODE,IW) + MDEL(NODE,IC)
      30        CONTINUE
                IF(MDELS(NODE,IW)) 25,26,26
      26        KTR=KTR+1
                K1(NODE,IW)=1
                K2(NODE,IW)=0
      25        CONTINUE
                DO 4386 IW=1,NW
                IF(K2(NODE,IW).EQ.0) GO TO 4386
                GO TO 43861
      4386      CONTINUE
                GO TO 789
      43861     IF(KTR.EQ.0) GO TO 789
      787       CONTINUE
                DO 41 IW=1,NW
                IF(K2(NODE,IW).EQ.0) GO TO 41
                LN(NODE,IW)=ILN(IW)
                JD(NODE,IW)=IJD(IW)
                DO 41 IC=1,NC
                MM=IDEL(NODE,IC)
                IF(K1(NODE,MM).EQ.0) GO TO 41
                LN(NODE,IW)=LN(NODE,IW)-1
                JD(NODE,IW)=JD(NODE,IW)-ID(IC)
      41        CONTINUE
                JW=1
      43        IF(K1(NODE,JW).EQ.1) GO TO 44
                JW=JW+1
                GO TO 43
      44        DO 45 IC=1,NC
      45        MINC(IC)=IVC(JW,IC)
                JW=JW+1
                IF(JW.GT.NW) GO TO 47
                DO 46 IW=JW,NW
```

```
               IF(K1(NODE,IW).EQ.0) GO TO 46
            DO 48 IC=1,NC
   48       MINC(IC)=AMIN1(MINC(IC),IVC(IW,IC))
   46       CONTINUE
   47       KTR=0
            DO 49 IW=1,NW
            IF(K2(NODE,IW).EQ.0) GO TO 49
            MEGAS(NODE,IW)=-IFC(IW)
            DO 50 IC=1,NC
            MEGAS(NODE,IW)=MEGAS(NODE,IW)+AMAX1(0.,MINC(IC)-IVC(IW,IC))
   50       CONTINUE
            IF(MEGAS(NODE,IW).GT.0.) GO TO 49
            KZ(NODE,IW)=1
            K2(NODE,IW)=0
            KTR=KTR+1
   49       CONTINUE
            DO 4329 IW=1,NW
            IF(K2(NODE,IW).EQ.0) GO TO 4329
            GO TO 43291
   4329     CONTINUE
            GO TO 789
   43291    IF(KTR) 789,789,786
   789      Z(NODE)=0.
            DO 60 IW=1,NW
            IF(K1(NODE,IW).EQ.1) GO TO 52
            Y(NODE,IW)=0.
            GO TO 60
   52       Y(NODE,IW)=1.
C
C           LINEAR PROGRAM
C
   60       CONTINUE
            DO 53 IC=1,NC
            KW=IDEL(NODE,IC)
            IF(KZ(NODE,KW).EQ.1) GO TO 538
            IF(K1(NODE,KW).EQ.1)GO TO 54
            IF(LN(NODE,KW).EQ.0) XX=9.999999E 50
            IF(LN(NODE,KW).EQ.0) GO TO 151
            XJN=FLOAT(LN(NODE,KW))
            XX=IFC(KW)/XJN
   151      IF(MDEL(NODE,IC).GT.XX)  GO TO 54
   538      JW=1
   540      IF(KZ(NODE,JW).EQ.0) GO TO 539
            JW=JW+1
            GO TO 540
   539      AA=IVC(JW,IC)
            IF(LN(NODE,JW).EQ.0) XX=9.999999E 50
            IF(LN(NODE,JW).EQ.0) GO TO 152
```

```
            XJN=FLOAT(LN(NODE,JW))
            XX=IFC(JW)/XJN
      152   IF(K2(NODE,JW).EQ.1) AA=AA + XX
            KW=JW
            JW=JW+1
            IF(JW.GT.NW) GO TO 54
            DO 55 IW=JW,NW
            IF(K2(NODE,IW).EQ.1) GO TO 55
            BB=IVC(IW,IC)
            IF(LN(NODE,IW).EQ.0) XX=9.999999E 50
            IF(LN(NODE,IW).EQ.0) GO TO 153
            XJN=FLOAT(LN(NODE,IW))
            XX=IFC(IW)/XJN
      153   IF(K2(NODE,IW).EQ.1)BB=BB + XX
            IF(AA.NE.BB) GO TO 56
            IF(K1(NODE,IW).EQ.1) GO TO 57
            GO TO 55
      56    AA=AMIN1(AA,BB)
            IF(AA.NE.BB) GO TO 55
      57    KW=IW
      55    CONTINUE
      54    XJN=FLOAT(LN(NODE,KW))
            IF(LN(NODE,KW).EQ.0) XX=9.999999E 50
            IF(LN(NODE,KW).EQ.0) GO TO 154
            XX=1./XJN
      154   IF(K1(NODE,KW).EQ.1) GO TO 58
            Y(NODE,KW)=XX + Y(NODE,KW)
      58    Z(NODE)=Z(NODE)+IVC(KW,IC)
      53    ISOL(NODE,IC)=KW
            KTR=0
            DO 4173 IW=1,NW
            IF(Y(NODE,IW).EQ.0.) GO TO 4174
            Z(NODE)=Z(NODE)+IFC(IW)*Y(NODE,IW)
      4174  IF(Y(NODE,IW).EQ.0..OR.Y(NODE,IW).EQ.1.) GO TO 4173
            KTR=KTR+1
      4173  CONTINUE
            IF(KTR) 71,71,72
      71    CONTINUE
      C
      C     IS THE SOLUTION TERMINAL
      C
            IF(NFIRST.EQ.1) GO TO 711
            WRITE(6,5833) Z(NODE),ITER
            NFIRST=1
      C
      C     IS THE SOLUTION OPTIMAL
      C
      711   IF(NODE.EQ.1) GO TO 6789
```

```
              IF(UBD.GT.Z(NODE)) GO TO 790
              Z(NODE)=LLN
              KODE=NODE
              IF(NKTR.NE.0.OR.NKTR1.NE.0) GO TO 1
              GO TO 1236
      790     UBD=Z(NODE)
              IF(NUBDN.NE.1) KODE=NUBDN
              NUBDN=NODE
              ITROPT=ITER
              Z(NODE)=LLN
              IF(NKTR.NE.0.OR.NKTR1.NE.0) GO TO 1
      1236    CONTINUE
              JW=1
              XLN=LLN
      7911    JW=JW+1
              IF(Z(JW).LT.UBD) GO TO 7910
              IF(Z(JW).LT.LLN) KODE=JW
              Z(JW)=LLN
      C
      C       IS THE SOLUTION OPTIMAL
      C
              IF(JW-MODE) 7911,7789,7789
      7910    XLBD=Z(JW)
              NLBDN=JW
              IF(JW.EQ.MODE) GO TO 7914
              JW=JW+1
              DO 7913 I=JW,MODE
              IF(Z(I).LT.UBD) GO TO 77913
              IF(Z(I).LT.LLN) KODE=I
              Z(I)=LLN
              GO TO 7913
      77913   IF(XLBD.LE.Z(I)) GO TO 7913
              XLBD=Z(I)
              NLBDN=I
      7913    CONTINUE
      7914    CONTINUE
      C
      C       IS THE SOLUTION OPTIMAL
      C
              IF(UBD.LE.XLBD) GO TO 7789
              Z(NLBDN)=LLN
              GO TO 2163
      72      IF(NODE.NE.1) GO TO 791
              XLBD=Z(NODE)
              NLBDN=NODE
              Z(NODE)=LLN
              GO TO 2163
      74      Z(NODE)=LLN
```

```
              KODE=NODE
              GO TO 7791
      791     IF(Z(NODE).LT.UBD) GO TO 7791
              Z(NODE)=LLN
              KODE=NODE
      7791    IF(NKTR.NE.0.OR.NKTR1.NE.0) GO TO 1
              GO TO 1236
      2163    JW=1
              NODE=NLBDN
      5781    IF(K2(NODE,JW).EQ.1) GO TO 5782
              JW=JW+1
              GO TO 5781
      5782    CONTINUE
      C
      C       A FREE OFFICE IS SELECTED BY A BRANCHING DECISION RULE
      C
              KKW=JW
              JW=JW+1
              DO 6721 I=JW,NW
              IF(K2(NODE,I).EQ.0) GO TO 6721
              IF(MEGAS(NODE,KKW).GE.MEGAS(NODE,I)) GO TO 6721
              KKW=I
      6721    CONTINUE
              NKTR1=2
              GO TO 1
      6789    UBD=Z(NODE)
              ITROPT=ITER
              GO TO 7789
      9779    WRITE(6,97791)
              GO TO 97792
      7789    CONTINUE
      97792   CONTINUE
              DO 83 I=1,NW
              IF(Y(NUBDN,I).EQ.0.) GO TO 83
              UBD=UBD - FWCOST
      83      CONTINUE
              WRITE(6,10004)ITER,UBD,ITROPT,MODE
              DO 82 I=1,NW
              FTE=0.
              IF(Y(NUBDN,I).EQ.0.) GO TO 82
               WRITE(6 ,10005) OFF(I),OFFF(I)
              DO 81 J=1,NC
              IF(ISOL(NUBDN,J).NE.I) GO TO 81
              FTE=FTE + ((ID(J)/XRATE)/DRPEMP)
              WRITE(6 ,10006) J,IVC(I,J)
      81      CONTINUE
              WRITE(6 ,10008) OFF(I),OFFF(I),FTE
      82      CONTINUE
      99      CONTINUE
              CALL EXIT
              END
```