END

FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD A126079

Georgia Institute

of

Technology

A MINIMAL TECHNOLOGY ROUTING SYSTEM

by

John J. Bartholdi, III
Loren K. Platzman
R. Lee Collins
William H. Warden, III

83-01

DTIC

MAR 2 5 1983

A

PRODUCTION and DISTRIBUTION RESEARCH CENTER

83 02 023 170

A MINIMAL TECHNOLOGY ROUTING SYSTEM

by

John J. Bartholdi, III
Loren K. Platzman
R. Lee Collins
William H. Warden, III

83-01

DTIC
ELECTE
S MAR 2 5 1983 D

A

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
1-404-894-2300

0. Introduction

This describes the successful implementation of a novel routing system based on a new travelling salesman heuristic by the first two authors. The application called for the efficient daily routing of a varying number of vehicles to more than 200 delivery points whose locations change at a rate of about 14% each month. The system had to be easily maintained by one person and require no resources (e.g., no computer). *This* system achieved all of these objectives, cost less than $50, and, moreover, shortened average travel times by 13% compared to previous performance.

1. The Problem

A. The Organization

Senior Citizen Services, Inc. is a private, non-profit corporation in Atlanta, Georgia whose purpose is to provide social services for the elderly, especially the elderly poor, in Fulton County. One of their major services is the "Meals-on-Wheels" program (MOW), which delivers prepared lunchs to people who are unable to shop or cook for themselves. As for many charitable organizations, the funding for MOW is unstable, chronically insufficient, and occasionally desperate. Any additional resources are used to purchase more food for needy people, so the administrative facilities of MOW remain the bare minimum necessary to function.

B. The System

MOW operates Monday through Friday each week. At 8:30 a.m. the prepared meals are delivered by an institutional caterer to Senior Citizen Services in mid-Atlanta. There they are heated in a holding oven until about 9:15 a.m. when 4 paid, part-time employees arrive. They load the meals into insulated bulk containers and then into their 4 vehicles. Each driver is given a "route manifest" which lists all of his delivery loca-

tions in the suggested order of visitation. Each then drives his route, delivering 40-50 meals to 30-40 locations during 10 a.m. - 2 p.m.

C. Constraints

Neither of the constraints commonly associated with routing problems holds for this system. Since the delivery vehicles are usually station wagons, they can easily carry sufficient meals, so that vehicle capacity is not an effective limitation. The only time constraint is that all meals must be delivered within four hours, which is the time during which the insulated containers will keep the meals properly warm. However, since each driver completes his route within the limit, time constraints are usually not active. In fact, neither vehicle capacity nor delivery time is likely to be an active constraint unless the system were to grow considerably, an unlikely event for a charitable organization during lean times.

D. The Manager

MOW is managed by a devoted, energetic woman who is a full-time employee of Senior Citizen Services. Again, as in many charitable organizations, the manager tends to be overworked. Her responsibilities are many, and include

1. management of the MOW budget

2. responsibility for all technical aspects of meals

    a. participating in menu planning

    b. ordering of meals from caterer

    c. monitoring quality of meals

    d. maintaining insulated containers for the meals

    e. supervising the part-time employee who packs meals into insulated containers

2

3. management of delivery

    a. maintaining list of delivery locations

    b. maintaining routes to be followed by delivery trucks

    c. supervising the 4 part-time drivers.

4. recruiting and training volunteers

5. coordinating with social workers.

The manager has little time (and essentially no resources) to devote to routing.

E. Special Features

MOW maintains two lists of clients: an _active_ list of those to whom meals are currently delivered, and a _waiting_ list of those hoping to join the system when space or additional resources become available. A special feature of this delivery problem is that the lists are quite volatile. In fact, the lists change at a rate of about 14% each month. This is due to the nature of the clients: most are elderly and/or ill. They may die, or recover from illness, or receive care elsewhere (such as a hospital, nursing home, or family) and so leave whichever list they are on. Clients may be added to the active list either from the waiting list, or as emergency special cases (perhaps referred to MOW by a social worker).

The volatility of the active list is further increased by the special way in which MOW is funded. Senior Citizen Services receives operating revenues from three primary sources, the federal government, the state of Georgia, and United Way. Unfortunately, all three administer their grants under different fiscal calendars. The federal government begins its fiscal year on October 1, the State of Georgia on July 1, and United Way on January 1. The multiple fiscal years cause continuous turmoil at MOW

3

because each grant must be spent during its respective fiscal year. Consequently it is not unusual for a large number of people to be added to the active list during the close of a fiscal year and then to be removed to the waiting list during the beginning of a new fiscal year.

Table 1 shows changes to the active list during the last year.

F. Our Task

We set out to design a method to help a busy manager quickly generate efficient routes from a volatile list. This method could not rely on a computer, or even on an appreciable clerical effort, for such resources are not within the means of MOW.

2. A New Travelling Salesman Heuristic

The routing system we implemented is based on a new travelling salesman heuristic due to the first two authors. This heuristic is extremely simple and yet provides good tours on the average. The idea behind the algorithm is the "spacefilling curve" of Figure 1, which may be imagined to order all the points of the unit square according to the sequence in which they appear on the curve. Thus, to each location (x,y) on the square, there corresponds some real number θ $(0 < \theta < 1)$, where θ gives the relative location of the point (x,y) along the spacefilling curve. The algorithm has the following structure:

Step 1. For each location (x,y) calculate the corresponding θ

Step 2. Sort the locations according to θ.

Thus the points to be visited are sequenced according to the order in which they appear along the spacefilling curve, as illustrated in Figure 2.

We have shown that this heuristic generates tours that are about 25% longer than optimum (on the average, for random point sets). The quality

4

of this solution is competitive with that of the nearest neighbor heuristic as reported by Bentley and Saxe [1980]. However, several novel features of this new heuristic make it especially attractive. First we observe that it requires minimal data: to route to n locations requires only the 2n values of the coordinates (x,y). In fact, the $O(n^2)$ distances between points are never required! Second, we note that the algorithm is unusually fast (an order of magnitude faster than the nearest neighbor heuristic). Step 1 consists of evaluating a simple recursive equation, and requires only $O(k)$ steps if x and y are both given to k decimal places. (Details of this calculation as well as a complete study of the heuristic can be found in Bartholdi and Platzman [1982].) Step 2 is simply a sorting routine and may be accomplished in only $O(n \log n)$ time. This is the dominant step of the algorithm.

Since the algorithm is essentially sorting, other useful properties pertain. Most important for our purposes is that points may be easily inserted into or deleted from the heuristic tour. To insert a point, one merely computes its corresponding θ and inserts it into the sorted list of θ's with $O(\log n)$ effort. To delete a point, one locates it in the sorted list, again with $O(\log n)$ effort, and simple deletes it. There is no need to re-solve the entire problem. In contrast, tours generated by other methods are not so easy to modify. For example, after insertion/deletion from an optimal (or nearest neighbor) tour, considerably more work is required to recompute an optimal (or nearest neighbor) tour on the altered set of points. It is the minimal requirements of this algorithm together with the ease of insertion/deletion that suit it so well to the MOW problem.

5

Note the flexibility this route-partitioning gives to the manager. If, for example, a driver or vehicle is unavailable, it is simple to partition the route list into three sets of cards, thereby immediately determining three routes. We expect that this flexibility will be even more important for other MOW programs wherein the drivers are unpaid volunteers whose number varies daily.

The card files also permit easy insertion and deletion so that the system can easily handle the volatility of the lists. To delete a client, one simply looks up his name in the alphabetical list, notes his θ value, and removes the card; then one looks up his θ value in the route list and removes that card also. To insert a client, one simply goes to the map and measures the (x,y) coordinates of his location; one then enters the table with (x,y) and reads the corresponding θ value. After two identical file cards are prepared for the client, one is inserted into the alphabetical list and one is inserted into the route list (according to θ). (See Figures 5 and 6.)

4. Implementation, Operation, and Performance

We had some initial difficulty in implementing the system because the drivers did not want to change their routes. Each was familiar with his area of the city and general sequence of locations. Moreover, there was concern that too much change would upset the clients. Most of the clients are old, sick, and isolated, and for them the regular visit of a familiar driver is an important part of their day. However, because of budget reductions in July 1982, MOW had to severely reduce its active list and restructure the routes accordingly. Since major changes had to be made anyway, we implemented our system at that time.

7

Since the routing algorithm is a heuristic, we expected it to occasionally choose sequences that could easily be improved. Accordingly we advised drivers to consider their sequence a suggestion to which they should make local improvements if possible. In fact no clear improvements were found and we had more trouble with drivers erroneously thinking they knew an improvement to the sequence. Afterwards we strengthened our recommendation of the sequence exactly as determined by the heuristic.

We discovered that the partitioning of the single route into subroutes caused some concern. The manager naturally wanted "ideal" routes, i.e. those that look efficient on a map and balance total delivery time among the drivers. She was bothered that the routes derived by partitioning tend to have somewhat large travel times to each first or from each last delivery. It would have seemed more reasonable, she felt, to go someplace close first. But when we measured the routes, the relatively long initial and final legs of each were generally found to be an insignificant part of total delivery time. This was partly because the heuristic reduced total driving time to only 30-40% of total delivery time for each route. Thus occasional travel time aberrations from route partitioning tend to not matter. However, the MOW manager preferred to hunt and make adjustments until she determined an acceptable partition. Since repartitioning is done only occasionally, this seemed satisfactory.

The routing system is generally used as expected, with one exception. In practice, the manager tends to make additions to the active list by "eyeballing" the map, and not by looking up θ immediately. This is because daily changes are small and because the manager is rushed. Later, when time permits, she looks up the θ values and files them appropriately.

8

We note the danger of our system degenerating if the θ values are not maintained.

A goal of the manager is to balance the work of the four drivers. Since they are paid by the hour, none wants an unusually short route. Trading routes among the drivers is not acceptable because that would disrupt the driver-client relationship discussed earlier. Fortunately, the partitioning scheme works quite well in this regard. Since travel time is only 30-40% of total delivery time, the total delivery time depends mostly on how many meals are to be delivered. Consequently, partitioning the route file into equal sets of cards tends to produce routes for which the total delivery times (though not travel times) are nearly equal. Table 2 summarizes the routes as of October 1982.

Since the client list changed so quickly, it was not possible to directly compare the driving times and distances of our routes with previous routes. We did, however, submit a previous client list to our heuristic and determined that our routes were about 13% shorter as measured by euclidean distance.

The most important improvement, however, is the facility with which the system may now incorporate changes in the client list and/or the number of drivers.

5. Future Work

There are other programs similar to MOW in the several counties immediately surrounding Atlanta. Their problems are similar to those of MOW. While their active lists are smaller (typically half that of MOW) they are not able to afford paid deliverers. Consequently they are dependent on volunteer drivers, whose number may vary from day-to-day. We

expect a system like ours to help these organizations too. Accordingly, we are preparing a booklet describing how to implement such a system, giving θ tables, etc. This has the potential of helping the MOW-type organizations that are in most large cities.

It is evident that our method may be of use to various profitable delivery ventures as well, especially if it is not possible to justify a major investment in computer equipment and appropriately trained personnel. Potential application areas might include package ᵔarts, or newspaper deliveries.

References

J. L. Bentley and J. B. Saxe, "An Analysis of Two Heuristics for the Euclidean Travelling Salesman Problem," *Proceedings of the 18th Annual Allerton Conference on Communication, Control, and Computing*, 41-49, October 1980.

G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation Algorithms for Some Routing Problems," *SIAM J. Comput.*, Vol. 7, No. 2, May 1978.

J. J. Bartholdi, III and L. K. Platzman, "An O(n log n) Planar Travelling Salesman Heuristic Based on Spacefilling Curves," *Operations Research Letters*, Vol. 1, No. 4, September 1982, pp. 121-125.

Month	People Served	People Added	People Lost
July 1981	159	–	–
August	167	15	7
September	169	8	6
October	195	29	3
November	225	40	10
December	256	37	6
January 1982	?	?	?
February	273	31	?
March	299	41	15
April	327	36	8
May	341	26	12
June	353	22	10
July	358	5	20
August	278	2	62
September	240	2	40
October	227	8	21
November	246	40	21

Table 1: Changes in active list from August 1981 through October 1982
(Note: only about 80% of these clients received their meals)
through the routing system. Others are served by volunteers)
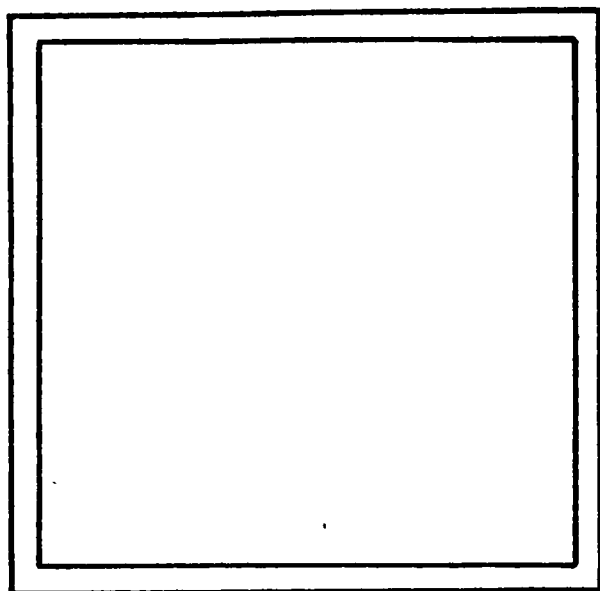
Curve$_1$
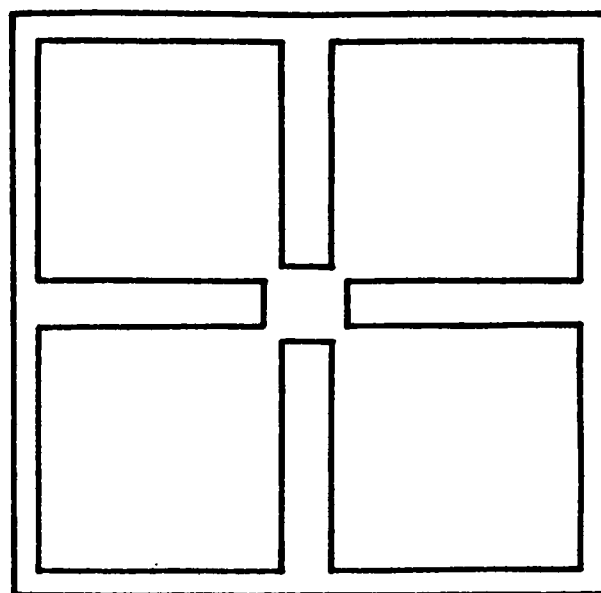
Curve$_2$
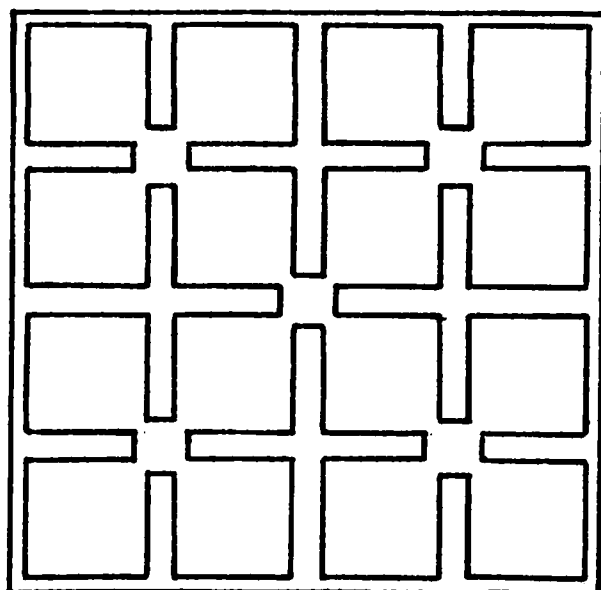
Curve$_3$
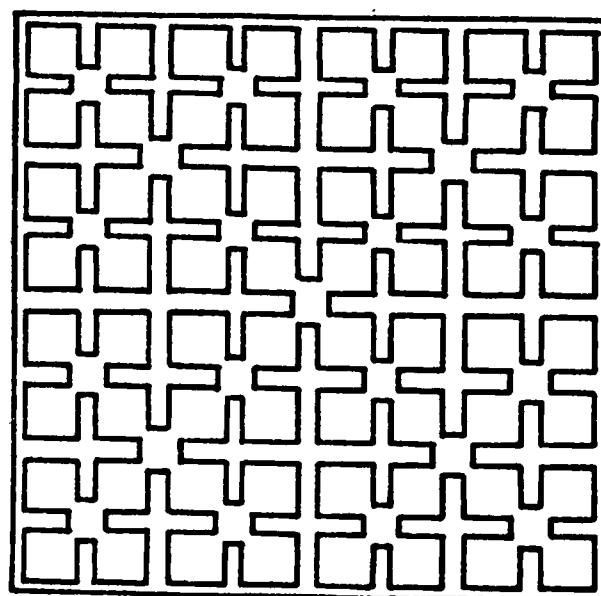
Curve$_4$

Figure 1: The spacefilling curve is the limit of the above sequence.
It passes through all of the points of the unit square.
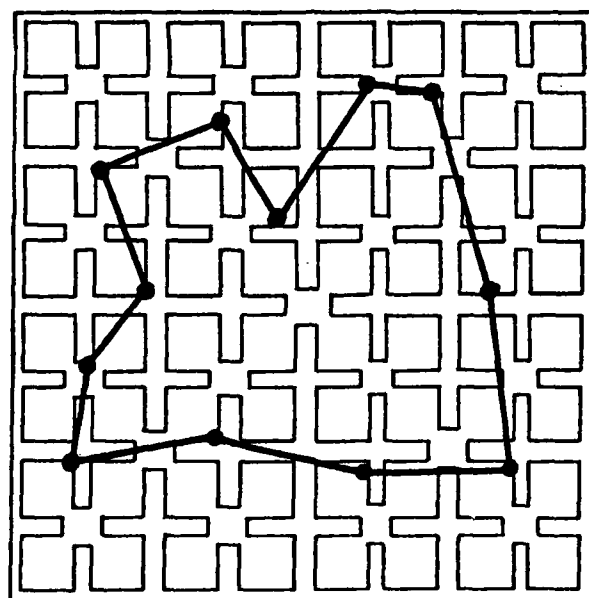
Figure 2: The heuristic tour visits the points in the order in which they appear along the spacefilling curve.
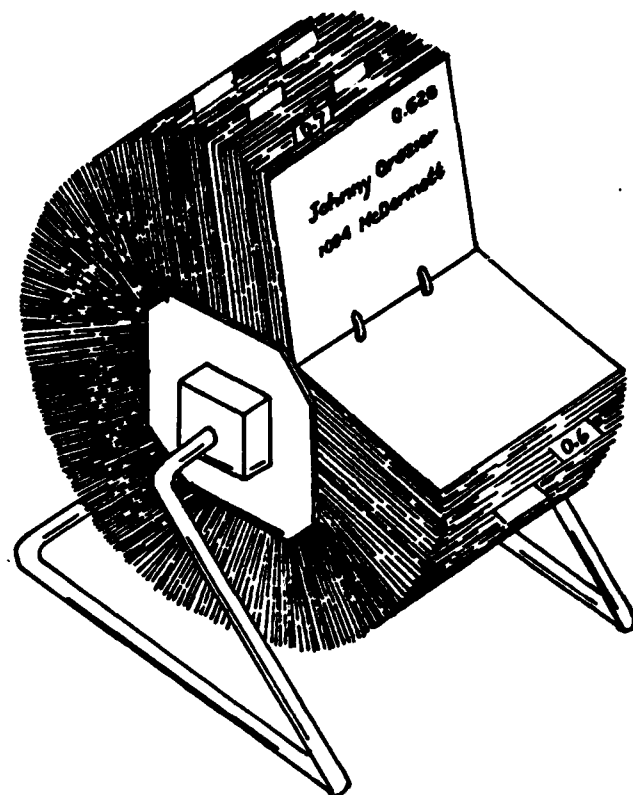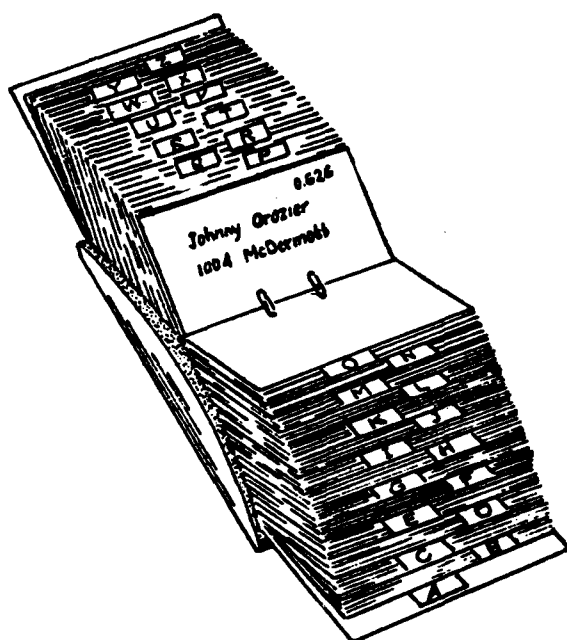
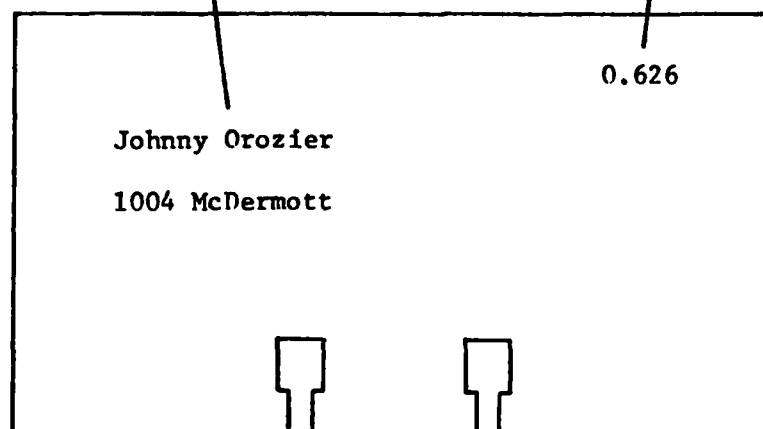sorted alphabetically -   - sorted by number

0.626

Johnny Orozier

1004 McDermott

Figure 3: The alphabetical file and the route file.
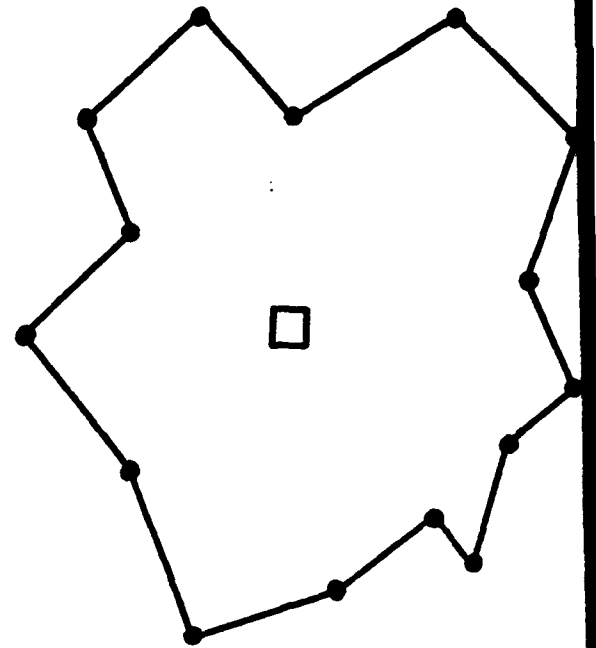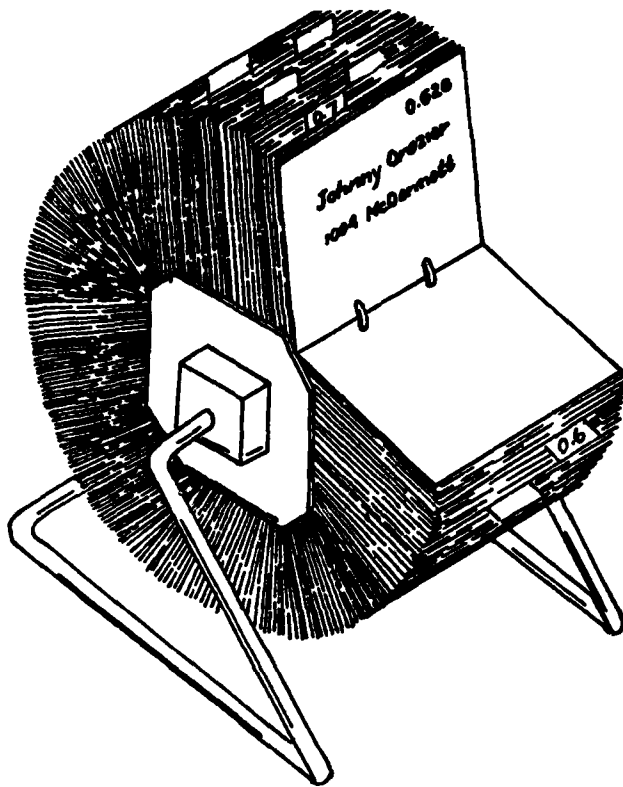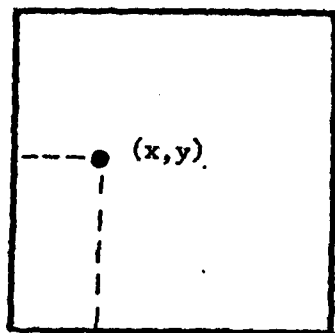
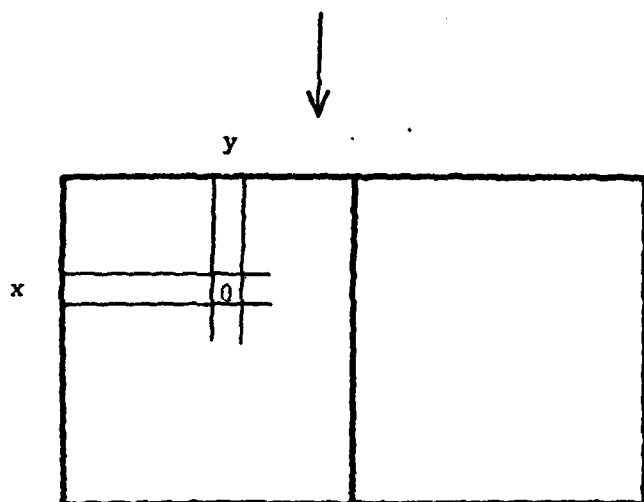Figure 4A: The sorted cards give an efficient route
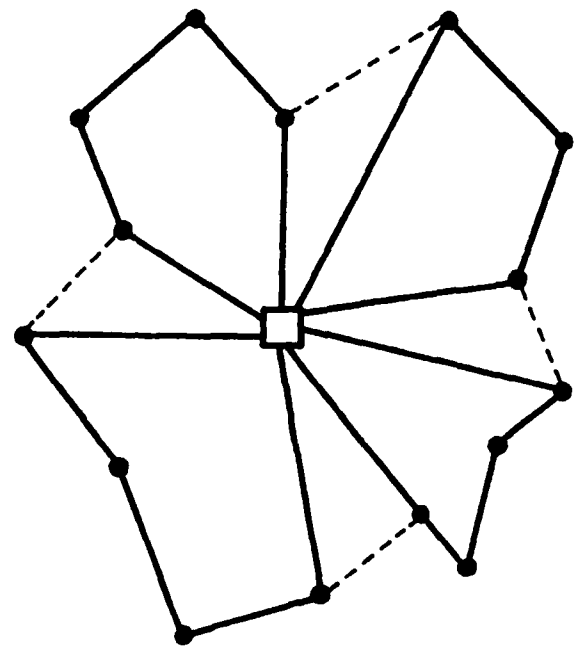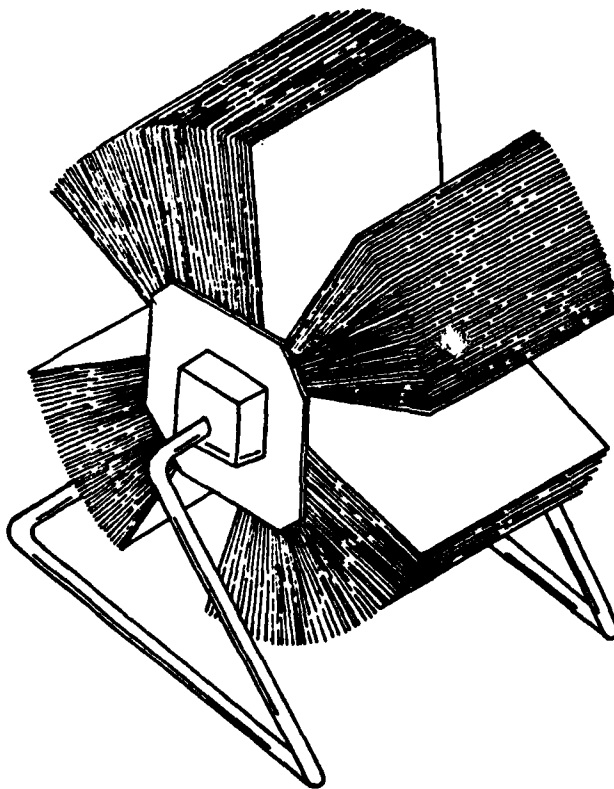
Figure 4B: Partitioning the cards gives efficient subroutes.

16

1. Find location on map and read (x,y) coordinates.

2. Enter table with (x,y) and read corresponding θ.

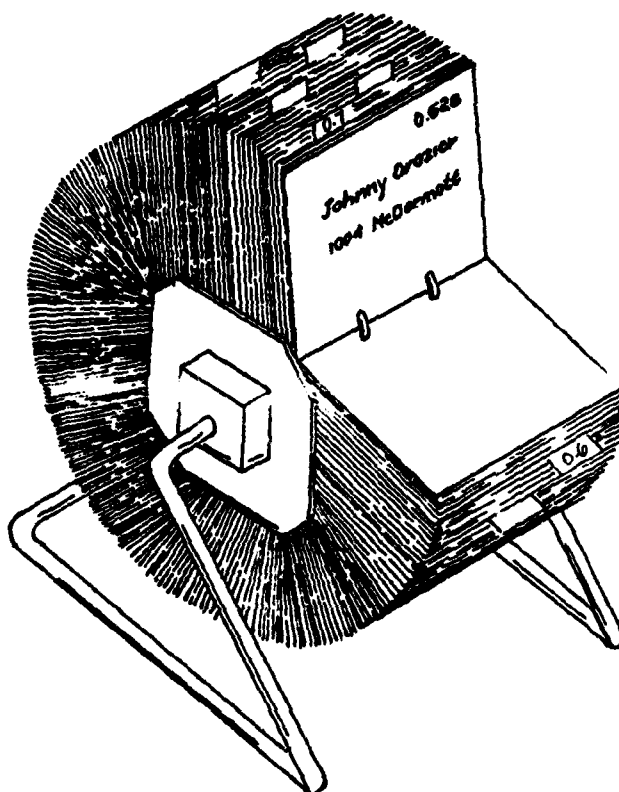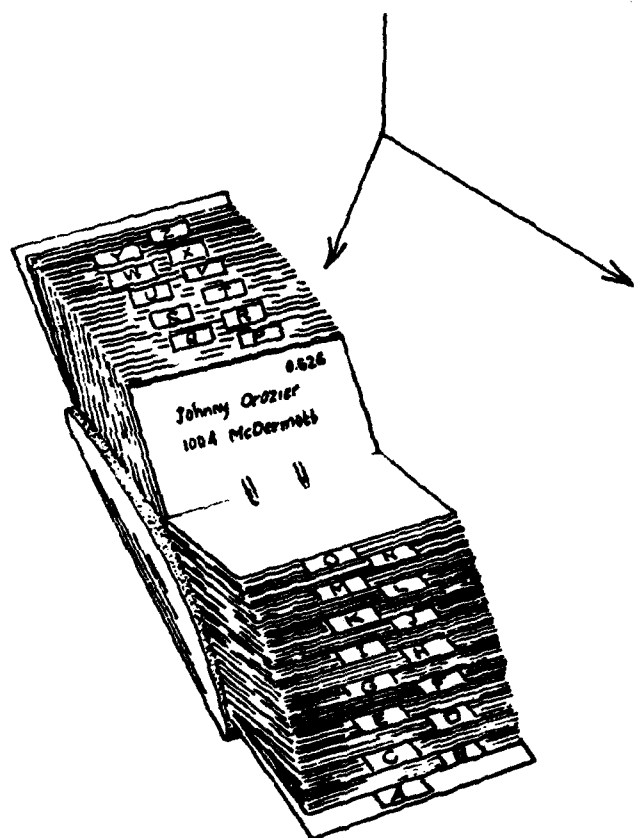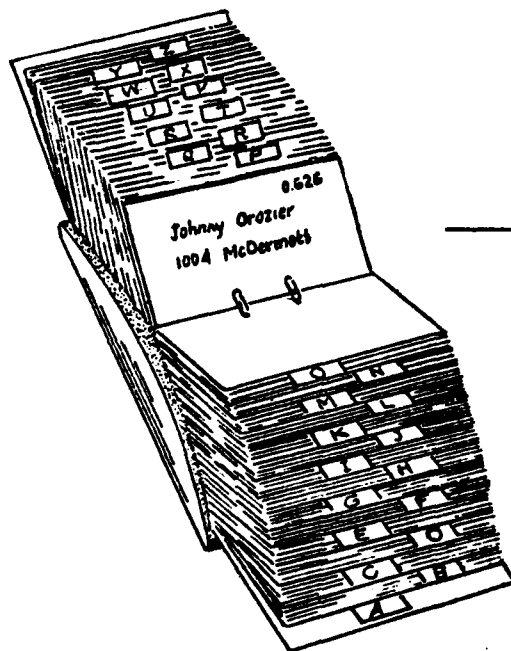3. Prepare two cards; insert one into the alphabetical list and one into the route list.

Figure 6: How to add a client to the system.

1. Find and remove card from the
   alphabetical list; note θ.

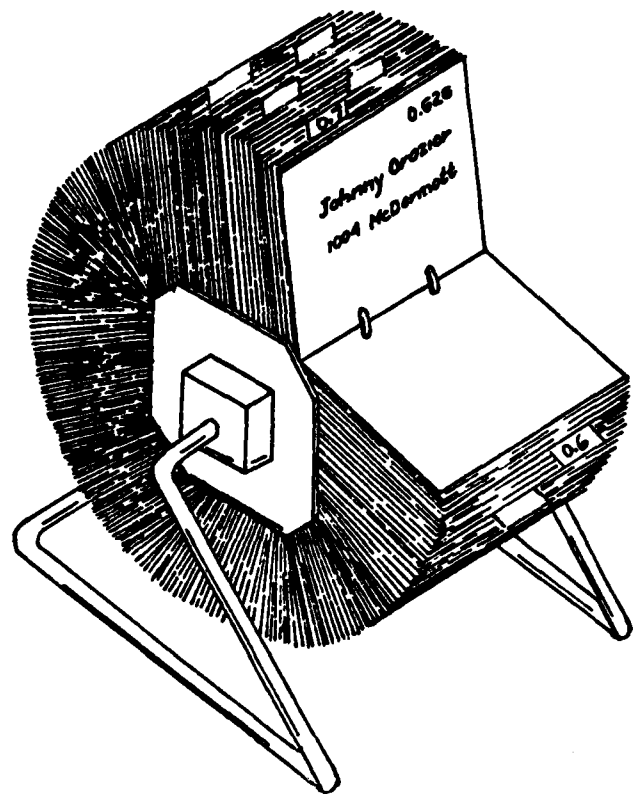2. Enter route list with θ;
   find and remove card.

Figure 5: How to remove a client from the system.

	Route 1	Route 2	Route 3	Route 4
Meals	48	47	55	45
Locations	25	35	38	36
Approximate miles	23	28	41	37
Approximate total delivery time (hours)	3 1/2	3 1/2	4	4

Table 2:   Summary of the routes as of November 1982