COMPUTERIZED OPTIMIZATION OF POSTAL ROUTES

by

VINUBHAI CHHOTABHAI PATEL

B.E. (M.E.), Maharaja Sayajirao University
Baroda, India, 1962

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1964

Approved by

[illegible]
Major Professor

LD
2668
R4
1964
P295
c.2

TABLE OF CONTENTS

INTRODUCTION

The postal route problem can be simply stated as the problem of selecting a route that will minimize the total distance travelled in visiting M blocks. The tour length can be expressed in feet, units of time, or other cost criteria. This problem is applicable to any city. It will require special consideration to treat this type of problem due to its symmetry.

This paper describes a technique (algorithm) and computer program which was used by the author to solve this problem. Though no effort has been made by the author of this paper to solve an assymmetric problem, enough published material is available which can be used to tackle such a type of problem when necessary.

This paper is organized in the following manner: An algorithm used to solve this problem is preceded by brief introduction of the problem. The second part contains some results of a particular problem. The computer program is given in the appendix.

HISTORY

At present a postman unaware of modern scientific planning, travels unnecessary additional distance every day and becomes unduly tired at the end of the routine monotonous job. If a postman has to cover a wider span of area than necessary the whole department suffers. This not only consumes the extra energy of the postman but also the precious time and money of the Post Office department. Instead, a well defined route for a postman may save a distance of a few hundred feet to be travelled by a postman.

The whole process of delivering the mail will be more efficient and effective than before by using scientific planning. The Post Office department can utilize services of postmen in a better way thereby making it more economical for the department.

The purpose of this paper is to find out the best possible route scientifically. This paper will explain the computerized algorithm used for this purpose.

A similar type of program developed by Mr. R. R. Gonzalez --Zubieta for "travelling salesman problem" can be used here. This program is now available at KSU IBM 1620 Computing Centre library but arrived too late for testing.

DEFINITION

The postal route problem is a specific case of the general problem of finding the smallest or shortest loop that connects a number of points in a plane. The distance between each point and every other point is usually recorded in a distance matrix. It should be noted that this distance matrix need not always be in terms of geometric distance since the optimum sequence or loop may be that which minimizes time or costs associated with the closed loop path. It is apparent then that time, costs or distance may be the variable of consideration.

The problem faced by the postman is to select a route that will minimize the total distance travelled while delivering the mail. If there are M points or blocks to be considered, then our problem is to choose a set of M elements, one in each row and one in each column, so as to minimize the sum of the elements chosen. Thus far the problem is identical with the assignment problem; however there are two further restrictions on our choice of element, one of which makes the problem considerably more difficult. Clearly we cannot choose an element along the diagonal, and this can be avoided by filling the diagonal with infinitely large elements. The second restriction is that we do not want a subloop in a route.

Before illustrating the problem with an example, we give a formal statement of the problem.

The postal route problem is a symmetrical problem. In other words distance to be travelled by a postman from point or block i to j is same as from j to i.

The distance to be travelled by a postman from block to block or from point to point forms a square matrix. The entry in row i and column j of the matrix is the distance from point i to point j and vice versa.

Stating in another way, if there are M points or blocks to be considered then it will give a square matrix of size MxM and with M_{ii}= infinity for each i. The problem will be to find a subset of M elements M_{pq}, M_{qr}, M_{rs}, ... M_{uv}, M_{vp} where p,q,... ,u,v are some permutation of integers 1,2, ...,M such that the sum of the elements of the subset shall be a minimum. Due to symmetry there are $\frac{1}{2}(M-1)!$ subsets or possible routes. Out of these there may be one or more routes which will give a minimum distance to be travelled by a postman.

PROCEDURE

The problem of minimizing the total distance travelled by a postman is solved by an algorithm. Basic method will be to break up the distance matrix into smaller and smaller size as the algorithm proceeds.

In this algorithm our first starting point will be the reduction of an original matrix. The smallest element of a row (which we will call as MINR in program) is to be subtracted from each element of the row and thus the row will be reduced. Then each column will be reduced by subtracting the smallest element of the column (MINC) from each element of the column. This process is continued for each row and column. This will

	1	2	3	4	5	6	7	8	9	10
1	∞	40	55	60	46	86	92	52	62	56
2	40	∞	15	20	5	45	51	12	21	16
3	55	15	∞	5	20	60	46	27	36	30
4	60	20	5	∞	15	55	61	21	31	36
5	46	5	20	15	∞	40	46	6	16	21
6	86	45	60	55	40	∞	6	46	55	60
7	92	51	46	61	46	6	∞	40	50	55
8	52	12	27	21	6	46	40	∞	10	15
9	62	21	36	31	16	55	50	10	∞	5
10	56	16	30	36	21	60	55	15	5	∞

Table 1

	1	2	3	4	5	6	7	8	9	10
1	∞	0 $\textcircled{6}$	15	20	6	46	52	11	22	16
2	0 $\textcircled{6}$	∞	10	15	0 $\textcircled{0}$	40	46	6	16	11
3	15	10	∞	0 $\textcircled{20}$	15	55	41	21	31	25
4	20	15	0 $\textcircled{20}$	∞	10	50	56	15	26	31
5	6	0 $\textcircled{0}$	15	10	∞	35	41	0 $\textcircled{4}$	11	16
6	45	39	34	49	34	∞	0 $\textcircled{68}$	39	49	54
7	51	45	40	55	40	0 $\textcircled{68}$	∞	33	44	49
8	11	6	21	15	0 $\textcircled{4}$	40	34	∞	4	9
9	22	16	31	26	11	50	45	4	∞	0 $\textcircled{13}$
10	16	11	25	31	16	55	50	9	0 $\textcircled{13}$	∞

Table 2

ensure a new matrix with non-negative elements and at least one zero in each row and column.

Table 1 is the original distance matrix of ten point problem. Reduction of the matrix by rows and then by columns gives the matrix of table 2.

We commence by treating the problem as an assignment problem. The zeros of this particular matrix (table2) provides a solution to the assignment problem, shown by distinguished elements. However, this is not a solution of the route problem, because it tells us to go from point 1 to 2 and then back to 1.

We examine the matrix for some of the "next best" solutions to the assignment problem, and try to find out one that satisfies the additional restrictions. It should be obvious from this matrix that any one of the cells which contains zero M(i,j) should be selected in going from X(X=i) to Y(Y=j).

We need to select the best available route which will give a minimum distance to be travelled. Since point j must connect to some point, the route must incur at least the cost of the smallest element in column j excluding M(i,j) and since point i must be reached from some point, the route must incur at least the cost of the smallest element in row i excluding M(i,j). The author has used MINEC and MINER for these minimum values in column and row excluding M(i,j). Sum of these two elements is denoted by L.

$$L = \text{MINER} + \text{MINEC}$$

We should calculate L for all cells which have zero value.

We shall select (X,Y) to be that pair that gives the largest L.

For example, the values L are written in small circles placed in the cells of zeros of table 2. The largest L is M(6,7) = MINER(6) = MINEC(7) = 34+34 = 68 and M(7,6) = MINER(7) + MINEC (6) = 33+35 = 68 and so either (6,7) or (7,6) will be the first pair to be selected. We know that due to symmetry each optimum route will give an optimum route in reverse direction also. Due to this we have to decide from the beginning which way to go and stick to it until final solution is reached. Due to symmetry there are chances that one will get two similar largest values of L. One of them will be always above the diagonal of matrix and other below the diagonal. In such circumstances select the largest L which is above the diagonal. Mathematically for two equal and largest values of L, select L for which j,i. In Fig.2 (6,7) satisfies our condition and is the point of our choice.

Since the pair (X,Y) is now committed to the tour, row X and column Y are no longer needed and are deleted from the matrix. For convenience, in the computer programming it was found easier to keep infinity (i.e. the largest number 9999) in rows and columns which were to be deleted. It will be noticed that (X,Y) now will be a part of connected path. Suppose the path starts at point f and ends at point g i.e. if we get a path (f,g), the connecting of g to f should be forbidden for it would create a subroute - a circuit with less than n points and no subroute can be a part of a final route.

This will be clear from Fig. 3. Row 6 and column 7 are deleted i.e. replaced by infinity and $N(7,6) = \infty$ should also be noticed.

$$
\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & \infty & 0 & 15 & 20 & 6 & 46 & \infty & 11 & 22 & 16 \\
2 & 0 & \infty & 10 & 15 & 0 & 40 & \infty & 6 & 16 & 11 \\
3 & 15 & 10 & \infty & 0 & 15 & 25 & \infty & 21 & 31 & 25 \\
4 & 20 & 15 & 0 & \infty & 10 & 50 & \infty & 15 & 26 & 31 \\
5 & 6 & 0 & 15 & 10 & \infty & 35 & \infty & 0 & 11 & 16 \\
6 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
7 & 51 & 45 & 40 & 55 & 40 & \infty & \infty & 33 & 44 & 49 \\
8 & 11 & 6 & 21 & 15 & 0 & 40 & \infty & \infty & 4 & 9 \\
9 & 22 & 16 & 31 & 26 & 11 & 50 & \infty & 4 & \infty & 0 \\
10 & 16 & 11 & 25 & 31 & 16 & 55 & \infty & 9 & 0 & \infty
\end{array}
$$

Fig. 3

Fig. 3 is now a new matrix which is modified. After these modifications the matrix can be reduced again. Check each row first and then each column for a zero element in it. If there is an absence of zero element in a row, subtract the minimum number $M(i,j)$ from each element of that row. Similarly after reducing all rows check the columns and repeat the whole process again. In Fig. 3 row 7 and column 6 do not have zero elements. In row 7 the minimum element is $M(7,8) = 33$. Subtract this amount from each element in the row 7. In column 6 the minimum element is M(5,6) = 35. Subtract this amount from each element in column 6. Now we have an entirely new modified matrix as shown in Fig. 4.

$$\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & \infty & 0\,\textcircled{6} & 15 & 20 & 6 & 11 & \infty & 11 & 22 & 16 \\
2 & 0\,\textcircled{6} & \infty & 10 & 15 & 0\,\textcircled{0} & 5 & \infty & 6 & 16 & 11 \\
3 & 15 & 10 & \infty & 0\,\textcircled{20} & 15 & 20 & \infty & 21 & 31 & 25 \\
4 & 20 & 15 & 0\,\textcircled{17} & \infty & 10 & 15 & \infty & 15 & 20 & 31 \\
5 & 6 & 0\,\textcircled{0} & 15 & 10 & \infty & 0\,\textcircled{5} & \infty & 0\,\textcircled{0} & 11 & 16 \\
6 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
7 & 18 & 12 & 7 & 22 & 7 & \infty & \infty & 0\,\textcircled{7} & 11 & 16 \\
8 & 11 & 6 & 21 & 15 & 0\,\textcircled{4} & 40 & \infty & \infty & 4 & 9 \\
9 & 22 & 16 & 31 & 26 & 11 & 50 & \infty & 4 & \infty & 0\,\textcircled{13} \\
10 & 16 & 11 & 25 & 31 & 16 & 20 & \infty & 9 & 0\,\textcircled{13} & \infty
\end{array}$$

Fig. 4

Having this new matrix find L for all M(i,j) = 0. Select the M(i,j) which has the maximum L. Values of L are written in small circles placed in the zero cells of Fig. 4. The largest L is 20 for M(3,4) and so (3,4) is second pair selected. As stated before, row 3 and column 4 are no longer needed and should be deleted from the matrix. Next substitute M(4,3) = ∞.

It may be worth giving another example of finding (g,f) to clarify a fundamental concept. Suppose the committed points or

Block pairs were (6,7) (7,8) (8,9) and (3,4) and (X,Y) were (7,8). Then connected route containing (X,Y) would start at 6 and end at 9 giving (g,f) = (9,6). Again we have similar confronting situation. Now we have to carry out routine calculations each time dealing with new matrix. Starting from the original matrix, rows and columns are deleted for the blocks or point pairs committed to the route, infinities are placed to block subroutes and at forbidden point pairs, and the resulting matrix is reduced.

By the time M is a 2x2 matrix, there are only two feasible (i,j) left and they complete the route.

If we examine algorithm closely it can be observed that crossing out of a row and column and blocking of the corresponding subroute creates a new postman route problem having one less point each time. The blocking of subroutes is a way of introducing the route restrictions into what is otherwise an assignment problem and is accomplished rather successfully by the algorithm.

The ten point problem used for illustration was fully solved on an IBM 1620 and is presented fully in detail in subsequent pages.

CALCULATIONS

Problems up to ten points can be solved easily by hand. Although the author has made no special study of the time required for hand calculations, experience is that a ten point problem can be solved in approximately an hour.

The principle testing of algorithm has been by machine on an

IBM 1620 and IBM 1410. The program was written in Fortran II to implement programming.

CONCLUSION

A scale map of a particular postal route is given in the appendix. The ten point problem used for illustration is a part of this particular route.

No change was found in present route followed by a postman. This proves the efficiency of the present route. The author would like to suggest a further study on routes in outlaying areas.

A very interesting case happened when the author was testing a twenty point route. The optimized route included all twenty points giving a minimum distance to be travelled to reach all the points, but this particular route omitted one block. This type of situation is undesirable. It is advisable in such cases to include one more point in the problem by adding an extra point in the middle of the missed block and this will be a new data to solve the problem.

The problem up to 15 points usually requires only a few minutes. The time grows exponentially, however, and for 30 points time increases appreciably on IBM 1620. As a rule of thumb, adding ten points to the problem multiplies time by a factor of three.

Curves (1) and (2) of the graph on page 18 shows the results of problems solved on IBM 1620 and IBM 1410.

ILLUSTRATION

```
    DATA

  10

9999004000550060004600860092005200620056
  409999001500200005004500510012002100 16
  5500159999000500200060004600270036 0030
  6000200005999900150055006100210031 0036
  4600050020001599990040004600060016 0021
  8600450060005500409999000600460055 0060
  9200510046006100460006999900400050 0055
  5200120027002100060046004099990010 0015
  6200210036003100160055005000109999 0005
  5600160030003600210060005500150005 9999


    REDUCED   MATRIX

9999   000150020000600460052001100220016
    99990010001    000400046000600160011
  1500109999   0001500550041002100310025
  200015   099990010005000056001500260031
   6   000150010999900350041   000110016
  4500390034004900349999   0003900490054
  510045004000550040   09999003300440049
  11000600210015   000400034999900040009
  2200160031002600110050004500049999   0
  160011002500310016005500500009   09999
```

```
9999   0  15  20   6  469999  11  22  16
   9999  10  15   0  409999   6  16  11
  15  109999   0  15  559999  21  31  25
  20  15   09999  10  509999  15  26  31
   6   0  15  109999  359999   0  11  16
9999999999999999999999999999999999999999
  51  45  40  55  4099999999  33  44  49
  11   6  21  15   0  4099999999   4   9
  22  16  31  26  11  509999   49999   0
  16  11  25  31  16  559999   9   09999
   6       7       68
```

```
9999   0  159999   6  119999  11  22  16
   9999  109999   0   59999   6  16  11
9999999999999999999999999999999999999999
  20  1599999999  10  159999  15  26  31
   6   0  1599999999   09999   0  11  16
9999999999999999999999999999999999999999
  18  12   79999   799999999   0  11  16
  11   6  219999   0   599999999   4   9
  22  16  319999  11  159999   49999   0
  16  11  259999  16  209999   9   09999
   3       4       20
```

```
9999    0    89999    6   119999   11   229999
     9999    39999    0    59999    6   169999
9999999999999999999999999999999999999999
   10    599999999    0    59999    5   169999
    6    0    899999999   09999    0   119999
9999999999999999999999999999999999999999
   18   12    09999    799999999    0   119999
   11    6   149999    0    599999999    49999
9999999999999999999999999999999999999999
   16   11   189999   16   209999    999999999
    9         10          13
```

```
9999    0    89999    6   119999   1199999999
     9999    39999    0    59999    699999999
9999999999999999999999999999999999999999
   10    599999999    0    59999    599999999
    6    0    899999999   09999    099999999
9999999999999999999999999999999999999999
   18   12    09999    799999999    099999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
    7    2    99999    7   11999999999999999
    8         9           7
```

```
9999999999999999999999999999999999999999
99999999   39999   0   59999   699999999
9999999999999999999999999999999999999999
  10999999999999   0   59999   599999999
   69999   899999999   09999   099999999
9999999999999999999999999999999999999999
  189999   09999   799999999   099999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
   59999   79999   5   99999999999999999
   1        2        6
```

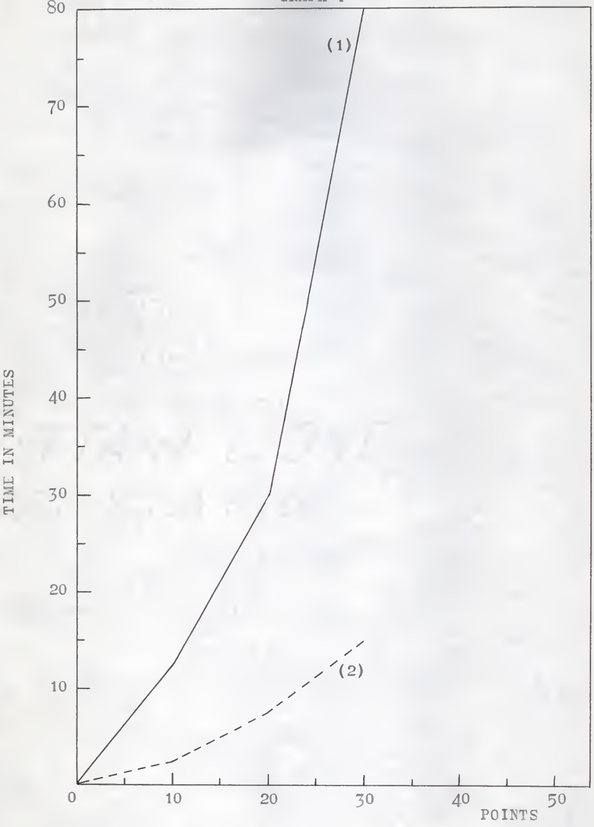
```
9999999999999999999999999999999999999999
99999999   39999   0   59999999999999999
9999999999999999999999999999999999999999
9999999999999999   0   59999   599999999
99999999   899999999   09999   099999999
9999999999999999999999999999999999999999
99999999   09999   799999999   099999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
  10        1        6
```

```
9999999999999999999999999999999999999999
99999999   39999   0999999999999999999999
9999999999999999999999999999999999999999
9999999999999999   099999999   599999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
99999999   09999999999999999   099999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
   5        6        5


9999999999999999999999999999999999999999
99999999   39999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999   0999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
   7        8        5
```

```
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
9999999999999999999999999999999999999999
   4        5
```

GRAPH-1



(1) Shows time required in minutes on IBM 1620

(2) Shows time reauired in minutes on IBM 1410

ACKNOWLEDGEMENTS

The author gratefully acknowledges the invaluable guidance rendered by the faculty and staff members in the Department of Industrial Engineering. In particular, the author wishes to express his sincere gratitude to his major professor and Director of IBM 1620 Computing Center at KSU, Professor J. J. Smaltz for suggestions, guidance, kind advice and encouragement during the entire study period. Thanks are also expressed to Mr. Larry Hofman for his excellent assistance in programming.

The assistance of the personnel at Post Office, Manhattan, Kansas in cooperating with the project is herewith gratefully accepted.

REFERENCES

1. Flood, M. M.
"The Traveling Salesman Problem", Opns. Res. 4, pp. 61-75 (1956)

2. Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M.
"Solution of a Large Scale Traveling Salesman Problem", Opns. Res. 2, pp. 393-410 (1954).

3. Little, John D. C., Murty, K. G., Sweeney, D. W., and Karel, Caroline
"An Algorithn for the Traveling Salesman Problem", Opns. Res. 6, pp. 972-989 (1963).

4. Sasieni, M., Yaspan, A., and Friedman, L.
"Operations Research", Sequencing. pp. 264-267.

5. Armbruster, J.
"The Traveling Salesman Problem", Unpublished paper May 12, 1962. Industrial Engineering Department, Kansas State University.

APPENDIX

MAIN   PROGRAM

```
              BCP  ROUTS PATL
      DIMENSION N(50,50)
      DIMENSION ICHEK(50),JCHEK(50)
      COMMON N
      IPRCT=0
      K=0
      KKK=0
      II=0
      JJ=0
      READ INPUT TAPE5,1,M
    1 FORMAT(I4)
      DO 101 I=1,M
  101 READ INPUT TAPE 5,2,(N(I,J),J=1,M)
    2 FORMAT (10I4)
    3 DO 6I=1,M
      L=MINR(I,M)
      IF(L-9999)4,6,4
    4 DO 6J=1,M
      IF(N(I,J)-9999)5,6,5
    5 N(I,J)=N(I,J)-L
    6 CONTINUE
```

```
   DO 7J=1,M
   L=MINC(J,M)
   IF(L-9999)30,7,30
30 DO 7I=1,M
   IF(N(I,J)-9999)31,7,31
31 N(I,J)=N(I,J)-L
 7 CONTINUE
   X1=-1
   X2=-1
   DO 8I=1,M
   DO 8 J=1,M
   IF (N(I,J))99,9,8
99 CALL EXIT
 9 K=J
   L1=MINER(I,M,J)
   L2=MINEC(J,M,I)
   X1=L1+L2
   IF(X1-X2)8,90,11
90 IF(J-I)8,8,11
11 X2=X1
   I1=I
   J1=J
 8 CONTINUE
   IPRCT=IPRCT+1
```

```
      ICHEK(IPRCT)=I1
      JCHEK(IPRCT)=J1
   12 FORMAT(2H0BI4,5X,I4,5X,I4)
      N(J1,I1)=9999
      DO 10 K=1,M
      N(I1,k)=9999
   10 N(K,J1)=9999
      DO 300 I=1,IPRCT
      DO 290 J=1,IPRCT
      IF(ICHEK(I)-JCHEK(J))290,280,290
  280 IROW=JCHEK(I)
      JCOL=ICHEK(J)
      N(IROW,JCOL)=9999
  281 DO 285 K=1,IPRCT
      IF(ICHEK(K)-IROW)285,282,285
  282 IROW=JCHEK(K)
      N(IROW,JCOL)=9999
      GO TO 281
  285 CONTINUE
      GO TO 300
  290 CONTINUE
  300 CONTINUE
      WRITE OUT PUT TAPE6,12,I1,J1,X2
      KKK=KKK+1
      IF(KKK-M)3,99,99
      END
```

SUBPROGRAM   MINR

```
                BCP  MINR
      FUNCTION MINR(I,M)
      DIMENSION N(50,50)
      COMMON N
      MINR=N(I,1)
      DO 100 J=2,M
      IF(N(I,J)-MINR)90,100,100
   90 MINR=N(I,J)
  100 CONTINUE
      RETURN
      END
```

SUBPROGRAM   MINC

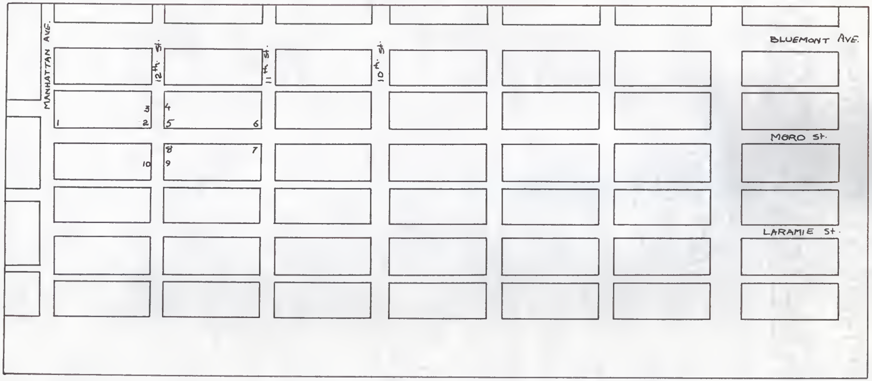
```
              BOP  MINC
      FUNCTION MINC(J,M)
      DIMENSION N(50,50)
      COMMON N
      MINC=N(1,J)
      DO 200 I=2,M
      IF(N(I,J)-MINC)190,200,200
  190 MINC=N(I,J)
  200  CONTINUE
      RETURN
      END
      SUBPROGRAM   MINER


               BCP  MINER
      FUNCTION MINER(I,M,K)
      DIMENSION N(50,50)
      COMMON N
      MINER=9999
      DO 10 J=1,M
      IF(J-K)5,10,5
    5 IF(MINER-N(I,J))10,10,8
    8 MINER=N(I,J)
   10 CONTINUE
      RETURN
      END
```

SUBPROGRAM   MINEC

```
            BOP  MINEC
      FUNCTION MINEC(J,M,K)
      DIMENSION N(50,50)
      COMMON N
      MINEC=9999
      DO 20 I=1,M
      IF(I-K)15,20,15
   15 IF(MINEC-N(I,J))20,20,18
   18 MINEC=N(I,J)
   20 CONTINUE
      RETURN
      END
```

MAP OF ILLUSTRATED ROUTE:



Scale: 1"=400 ft.

COMPUTERIZED OPTIMIZATION OF POSTAL ROUTES

by

VINUBHAI CHHOTABHAI PATEL

B.E. (M.E.), Maharaja Sayajirao University
Baroda, India, 1962

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1964

The purpose of this report was to study and find out some method (algorithm) which can be used to minimize the distance travelled by a postman while delivering mail. It was the purpose of this paper to computerize the algorithm so that the problem can be solved scientifically.

The problem of postal route was a symmetrical problem i.e. the distance to be travelled by a postman from block i to j is same as the distance from j to i. If there are M points or blocks to be considered then it will give a square matrix (distance matrix) of size MxM. The problem will be to find a subset of M elements N_{pq}, N_{qr}, ...N_{vp}, where p, q, ...u, v are some permutations of integers, 1, 2, ... M, such that the sum of the elements of the subset shall be a minimum. In matrix i stands for rows and j stands for columns.

The basic method will be to break up the distance matrix into smaller and smaller size as the algorithm proceeds.

The first step will be the reduction of the original matrix. Original distance matrix will be reduced by subtracting minimum element of row (i) from each elements of row, continuing this for all rows. Then subtract minimum element from each column (j). This will give a new reduced matrix having at least one zero element in each row and column.

The next step is to select from these elements, the best one which gives the minimum distance to be travelled. Take element N (X,Y) which has value zero. Select minimum amount from row X excluding N (X,Y). Select from column Y the minimum amount excluding N (X,Y). Sum these two elements and call it L. Similarly calculate L for all zero elements. Our required pair (X,Y) is that which gives the largest L.

Since the pair (X,Y) is now committed to the route, row X and column Y are no longer needed and should be deleted from the matrix. Now (X,Y) will be a part of a connected path. Suppose the path starts at point f and ends at point g i.e. if we get path (f,g), the connecting of g to f should be forbidden for it would create a subroute - a circuit with less than M points and no subroute can be a point of a final route. So (Y,X) should be substituted as infinity i.e. the largest possible number.

This process is continued till matrix is reduced to 2x2 matrix. By the time M is 2x2 matrix, there are only two feasible (i,j) left and they complete the route.

It can be observed that crossing out of the row and the column and the blocking of the corresponding subroute creates a new postman route problem having one fewer point each time.

The program was written in Fortran II for IBM 1620 to implement algorithm. The program was modified for IBM 1410.

Ten point problem solved by hand requires approximately an hour. The computer requires hardly few minutes to solve the problem. The time grows exponentially, however, and by 30 points is beginning to appreciable on IBM 1620. As a rule of thumb, adding ten points to the problem multiplies time by a factor of 3.

This technique can be more useful for rural areas and newly added areas.