BTAP: A COMPUTER PROGRAM TO OBTAIN SOLUTIONS TO THE BLOOD TRANSPORTATION-ALLOCATION PROBLEM AND OTHER TRAVELING SALESMAN TYPE PROBLEMS

ILHAN OR

and

WILLIAM P. PIERSKALLA

AUGUST, 1976

Department of Industrial Engineering
and Management Sciences
Northwestern University
Evanston, Illinois 60201

BTAP: A Computer Program to Obtain Solutions to the Blood Transportation-Allocation Problem and Other Traveling Salesman Type Problems

by

Ilhan Or

and

William P. Pierskalla

August, 1976

This research was partially supported by PHS-HS00786-01 from
the National Center for Health Services Research, and
by GK-40034 from the National Science Foundation,
William P. Pierskalla, Principal Investigator

This is a descriptive paper to supplement the research in [OR, 1976]. All of the new algorithms discussed in [OR, 1976] are coded in Fortran IV, to form the alternate library BTAP. This paper contains the actual codings and descriptions of the subroutines and functions in the library BTAP. Section 1 gives some general information on how to use BTAP. Section 2 contains the descriptions of the subroutines and of the important parameters and variables. The appendix contains the codings of the subroutines and functions in BTAP and of the main program BTP.

Section 1.

The library BTAP is designed to be used with the main program BTP. The input data for the package (BTAP and BTP) is read from two sources: TAPE5, which is usually a permanent file, and TAPE1, which is the keyboard. This package is built to run on a CDC 6400 computer and needs 75000 octal core memory and 55000 octal extended core storage to execute. TAPE5 should be composed of two files. The first file should contain the coordinates and the parameters γ_i of the hospitals involved in the given problem with the format "FORMAT (3F5.0)" (x-coordinate, y-coordinate, γ_i). Let there be N card images in this file. The second file should contain N card images with the format "FORMAT (I4, I6)" (each card image should contain a hospital and the amount of blood used per year in that hospital) and then values for SYCOST (1) to SYCOST (120) with the format "2X, 10I5".

Section 2.

The following is a short summary of all the important variables, parameters and other information in the program and how and where they are stored.

ALFA (150): For each hospital I, the expected number of emergency deliveries in one period, γ_i, is stored in ALFA (I). γ_i, i = 1,..., N are read and stored in subroutine READ1. ALFA (150) is in the common block |E|.

CANDID (600, 4): This is the candidate list. CANDID (I, 3), and CANDID (I, 4) contain the endpoint nodes of an edge. CANDID (I, 2) contain the node (hospital) "closest" to that edge. CANDID (I, 1) contains the "distance" of the closest node from that edge. CANDID (600, 4) is in the common block |D|.

DMT1 (150): This vector is used as temporary storage for information to be sent to or received from Extended Core Storage (ECS). If the ECS calling statement is "CALL READEC (DMT1, DATX (1, I), N)," DMT1 (J) will contain the Euclidean distance between nodes I and J. DMT 1 (150) is in common block |CM|.

DMT2 (150): This vector is used in the same way as DMT1. It is also in common block |CM|.

EMCOST (5): EMCOST (I) contains the total expected emergency delivery cost for region I. It is calculated in subroutine ALLOC1, ALLOC2, UPDATE and DISPACH. EMCOST (5) is in common block |K|.

IADRES (600): IADRES (I) contains the address (row number) of the I^{th} best candidate so far (i.e., the distance given by CANDID (IADRES (I), 1) is the I^{th} smallest) in the candidate list CANDID. It is computed in subroutine ADRES and used in subroutines BOUND1 and ALLOC2. It is in common block |D|.

IALTER (5, 5, 10): The vector IALTER (I, J, •) is equivalent to the set EX (I, J), which was defined and discussed in section 4.3 of [OR, 1976]. It is determined in subroutine IMPROVE and used in subroutines ALTER1 and ALTER2. IALTER (5, 5, 10) is in common block |F|.

IBANK (5): IBANK (I) contains the identification number of the central bank in region I. It is read from the keyboard in the main program BTP. IBANK (5) is in common block |C|.

IBLAD (150): IBLAD (I) contains the amount of blood used per year in hospital I. It is read from TAPE5 in subroutine READ1. IBLOOD (150) is in common block |E|.

IBLADTR (50): This array contains the number of units of blood carried by each periodic delivery vehicle. It is calculated and stored in subroutine DISPACH. IBLADTR (50) is in the common block |F|.

IBTR (5): IBTR (K) contains the number of active periodic delivery vehicles in region K. It is calculated and stored in subroutine DISPACH. IBTR (5) is in the common block |F|.

IHULL (300, 3): This array is used in the construction of tours in BOUND1 and ALLOC2. Each row contains information about an edge; the first two columns define the edge (they contain the two end nodes of the edge) and the third column indicates whether that edge is in the existing graph (value 0) or has been deleted (value 1). At the end of the subroutines BOUND1 and REFINE1, the first two columns of IHULL contain the resulting tours. IHULL (300, 3) is in common block |C|.

IHOSP (150, 5): IHOSP (I, 3) is a 0, 1 indicator used in subroutines BOUND1 and PFIND. IHOSP (I, 3) is zero when hospital "I" is not yet in the existing graph and one otherwise. IHOSP (I, 4) indicates the region hospital I is assigned to under the assignment algorithms ALLOC1, ALTER1 and ALTER2. IHOSP (I, 5) indicates the region hospital I is assigned to under the assignment algorithm ALLOC2. IHOSP (I, 1) and IHOSP (I, 2) indicate the two hospitals adjacent to hospital I in the latest routing obtained. IHOSP (150, 5) is in common block |A|.

IHULLA (150, 2): This array contains routing information in a form that can be used by the output subroutines, PLOTING and PRINTS. The first column of IHULLA contains the routing obtained using the assignments of subroutines ALLOC1 or ALTER1 or ALTER2 (of course, only the latest routing obtained is there, the earlier ones are erased). The second column of IHULLA contains the routings obtained using the assignments of subroutine ALLOC2. IHULLA (150, 2) is in common block |B|.

IIHOSP (150, 3): The first two columns of this array are used to store the original first two columns of IHOSP (150, 5), since the information contained in those columns of IHOSP is destroyed during the calls to BOUND1 and REFINE1 in subroutine TEST. IIHOSP (I, 1) and IIHOSP (I, 2) are used in subroutine ALTER2, in determining the two adjacent nodes to node $I \in EX(j_1, j_2)$ (see section 4.3 of OR [1976]). These two columns of IIHOSP are updated in subroutine UPDATE. The third column of IIHOSP is an indicator vector, IIHOSP (I, 3) = -1 if hospital I has never been in an exchange, IIHOSP (I, 3) = 0 if hospital I has already been permanently assigned to another region, IIHOSP (I, 3) = J if the pair (I, J) has been in an exchange before. This convention is helpful in avoiding duplicate tests in subroutine ALTER2. IIHOSP (150, 3) is in common block |F|.

N: N is the number of hospitals in the whole system. It is determined in subroutine READ1.

NALTER (5, 5): NALTER (I, J) contains the number of elements in the set EX (I, J) (or equivalently in the vector IALTER (I, J, •)). It is determined in subroutine IMPROVE and used in subroutines ALTER1 and ALTER2. NALTER (5, 5) is in common block |F|.

NBANK: This is the number of central blood banks in the whole system. It is read from the keyboard in the main program, BTP.

NUM (50): This vector contains the number of stops for each vehicle. It is in common block |C|.

NUMTR (50): NUMTR (I) contains the number of hospitals in region I. It is in common block |F|.

SCAL: It gives the scale of the plots. It is read from the keyboard in the main program BTP.

SYCOST (120): SYCOST (K) gives the systems cost estimate for a region containing K hospitals. It is read from TAPE5 in subroutine READ1. It is in common block |K|.

TSCOST (5): TSCOST (I) gives the total periodic delivery costs in region I. TSCOST (5) is in common block |K|.

X (150): X (I) contains the x-coordinate of hospital I. It is read from TAPE5 in subroutine READ1. X (150) is in common block |E|.

Y (150): Y (I) contains the y-coordinate of hospital I. It is read from TAPE5 in subroutine READ1. Y (150) is in common block |E|.

There are 19 subroutines and one function in the library BTAP (Blood Transportation-Allocation Problem). The following is a brief description of them.

SUBROUTINE READ1 (N)

The function of this subroutine is to read the input data from TAPE5. First X, Y coordinates and the parameter γ_i for each hospital are read and stored in vectors X (150), Y (150) and ALFA (150), respectively. Number of hospitals in the whole system, N, is determined. The blood usage of hospitals are read and stored in IBLAD (150). The system costs for all feasible system sizes are read and stored in SYCOST (120).

SUBROUTINE DISMAT1 (N)

The function of this subroutine is to calculate the distance matrix. For each pair of hospitals (I, J), the Euclidean distance between them is computed and stored in ECS (Extended Core Storage) at address DATX (I, J). Information storage to (and retrieval from) ECS is done one column at a time.

SUBROUTINE ALLOC1 (N, NBANK)

The function of this subroutine is to determine the hospital assignments that will minimize the expected emergency blood delivery costs (i.e., the assignments given by y^0 as defined in section 4.2 of OR [1976]). Thus, the closest bank J to each hospital I is determined. I is assigned to bank J by storing the group number of J in IHOSP (I, 4). Then, for each group K, the number of hospitals in it, the amount of blood used in it and the total expected emergency delivery costs for it are calculated and stored in NUM (J), NBLAD (J) and EMCOST (J), respectively.

SUBROUTINE ALLOC2 (N, NBANK)

This subroutine corresponds to the assignment algorithm described in section 3.5 of OR [1976]. The group number of each hospital I is determined (as described in section 3.5) and stored in IHOSP (I, 5). Then for each group, J, the number of hospitals in it, the amount of blood used in it, and the total expected emergency delivery costs for it are calculated and stored in NUM (J), NBLAD (J) and EMCOST (J), respectively.

SUBROUTINE TRAVEL (N, NBANK, KX)

Once the assignments of the hospitals (either y^0 given by ALLOC1 or y^* given by ALLOC2) are completed, this subroutine determines the routings for the MTSP (by calling the subroutines related to the convex hull algorithm) and stores them in column KX of IHULLA (150, 2) (column 1 if the given allocation is y^0 and column 2 if the given allocation is y^*) in the form that is required by subroutines PRINTS and PLOTING.

SUBROUTINE DISPACH (N, NBANK, KX, STOPS, UNITS, SCAL)

Once the assignments of the hospitals are completed (either y^0 given by ALLOC1 or y^* given by ALLOC2 or $\tilde{y}$ given by ALTER1 or ALTER2 or $\hat{y}$ supplied externally), this subroutine determines the routings for the MVDP (using the sweep algorithm discussed in section 2.3.4.3 of OR [1976]) and stores them in column KK of IHULLA (150, 2) in the form that is required by subroutines PRINTS and PLOTING (KK is 1 if the assignments are given by y^0, $\hat{y}$ or $\tilde{y}$ and KK is 2 if the assignments are given by y^*). Number of vehicles serving each region, IBTR (1) to IBTR (NBANK); the total number of vehicles, JSTAR; the number of stops for each vehicle, NUMTR (1) to NUMTR (JSTAR); the amount of blood carried by each vehicle, IBLADTR (1) to IBLADTR (JSTAR); the total expected emergency cost for each region, EMCOST (1) to EMCOST (NBANK); the total periodic delivery cost for each region, TSCOST (1) to TSCOST (NBANK); the total amount of blood used per year in each region, NBLAD (1) to NBLAD (NBANK) are also determined in the subroutine. (Note that the calls to subroutines CHULL1, BOUND1 and REFINE1 from this subroutine determine just single vehicle routes.)

SUBROUTINE CHULL1 (N, IC, KX, IX)

The function of this subroutine is to determine the graph, G, which is the boundary of the convex hull of the nodes in region IX. The edges in G are stored in IHULL (300, 3). Throughout the subroutine (in fact, throughout the whole program), "IHOSP (I, 4+KX)" is compared with "IX", to screen away the nodes that are not in region IX; in case of an inequality hospital I is disregarded. KX is a 0, 1 variable allowing the user to select either the fourth or the fifth column of IHOSP for screening. IC gives the number of edges in the existing graph, it is updated in subroutine PFIND.

The procedure used to determine G is as follows:

First the nodes with the smallest and the largest x-coordinates are determined (let them be H_{i1} and H_{ip}, respectively). Then, starting with

H_{i1}, the following step is applied until H_{ip} is reached: "Given H_i, find H_j such that the slope of the line passing from H_i, H_j is the maximum over all nodes that have a larger x-coordinate than H_i. Store H_j, let $H_i = H_j$." When H_{ip} is reached, the following step is applied until H_{i1} is reached: "Given H_i find H_j such that the slope of the line passing from H_i, H_j is the maximum over all nodes that have a smaller x-coordinate than H_i. Store H_j, let $H_i = H_j$." Once H_{i1} is reached, the graph G will be completely determined.

SUBROUTINE PFIND (N, NSIZE, K1, K2, KPOINT, KX, IX)

The function of this subroutine is to determine the "closest" node, IBEST, out of all the nodes in region IX that are not in the graph yet, to the edge defined by the end nodes K1 and K2. Then the "distance" from IBEST to edge (K1, K2), IBEST, K1 and K2 are stored in row NSIZE of the candidate list CANDID (600, 4). NSIZE which contains the total number of candidates (rows) in CANDID (600, 4) is updated in this subroutine. KPOINT is a pointer passed on to subroutine ADRES.

If this subroutine is entered through the calling statement CALL PFIND (which is the case in subroutines CHULL and BOUND1), the distance criterion used is "R $\times$ D" (Ratio $\times$ Difference - see section 3.4.2 of OR [1976] for detailed explanations). On the other hand, if this subroutine is entered through the calling statement CALL AFIND (which is the case in subroutine ALLOC2), the distance criterion used is "ANG $\times$ SH (see section 3.5 of OR [1976] for explanations).

SUBROUTINE ADRES (NSIZE, KPOINT)

The function of this subroutine is to order the candidates (rows) in the array CANDID according to their first attribute (the first attribute contains the "distance" of the candidate node, which is the second attribute, from the edge defined by the third and fourth attributes). The ordering is done by rearranging the entries of the adres vector to the candidate list, IADRES (600).

In theory, the best candidate (the row with the smallest value in column 1) is taken out of the candidate list and considered for insertion into the graph. In practice, however, it is much faster and easier not to delete anything from CANDID and IADRES, but instead to increase KPOINT, the pointer to the starting point in IADRES, by one unit. So at any instant in the algorithm, IADRES (1) to IADRES (KPOINT) contains the row numbers, in CANDID (600, 4), of candidates that have already been considered (i.e., the candidates that should have been deleted). Hence, IADRES (1) to IADRES (KPOINT) is disregarded and the rest of the vector IADRES is rearranged such that IADRES (KPOINT + I) contains the row number of the candidate node that is I^{th} closest to the existing graph.

SUBROUTINE BOUND1 (N, IC, COST, KX, IX)

The function of this subroutine is to modify G, the boundary of the convex hull of all nodes in group IX, into a circuit passing from all of the nodes in group IX. In other words, this subroutine executes step 3, 4, 5 and 6 of the convex hull algorithm (see section 3.4.2 of OR [1976]). Subroutine PFIND is called to find new candidates and to store and order them. The modified graph is stored in IHULL (300, 3). IC keeps track of the number of edges stored in IHULL. In theory, every time the existing graph is modified one edge is deleted and two new edges are added. In practice, however, no physical deletion from IHULL occurs. The third column of IHULL is considered to be a vector of flags, and every time the algorithm calls for a deletion the flag of the edge that is to be deleted is changed from 0 to 1.

Once the traveling salesman route for group IX is determined, the cost is computed and stored in COST (IX). Also the two adjacent nodes to each node I are stored in IHOSP (I, 1) and IHOSP (I, 2) and IHULL is rearranged so that columns 1 and 2 in the first NUM (IX) rows contain the resulting traveling salesman route.

SUBROUTINE REFINE1 (N, COST, KX, IX)

The function of this subroutine is to implement extension 2 of the convex hull algorithm (see section 3.4.4 of OR [1976]). If an improvement is found, subroutine CHANGE is called to make the change permanent. At the end of all the tests and changes, the resulting traveling salesman tour is stored in the first NUM (IX) rows of columns 1 and 2 of IHULL.

SUBROUTINE CHANGE (I1, I2, J1, J2, K1, K2, N)

The function of this subroutine is to modify the first two columns of IHOSP so that the path from I1 to I2 lying between nodes J1 and J2 in the existing graph would be placed between nodes K2 and K1.

SUBROUTINE IMPROVE (N, NBANK, SCAL, EXINDP)

The function of this subroutine is to determine and order the sets EX (j_1, j_2), $j_1, j_2 = 1, \ldots,$ NBANK (see section 4.3 of OR [1976] for the description of EX (j_1, j_2)). NALTER $(j_1, j_2, \cdot)$ correspond to set EX (j_1, j_2), and can have at most 10 elements. The routings given by x^* are saved in the first two columns of IIHOSP (150, 3). EXINDP is a logical parameter provided by the user from the keyboard that is used in deciding between algorithm 1 and algorithm 2 (see section 4.3 of OR [1976] for the descriptions of these algorithms). If EXINDP = .T., subroutine ALTER1 is called and algorithm 1 is applied; if EXINDP = .F., subroutine ALTER2 is called and algorithm 2 is applied.

SUBROUTINE ALTER1 (N, NBANK, SCAL)

This subroutine corresponds to step 9 of algorithm 1 for the MTSP. The node that is to be temporarily placed in a different region is stored in LIST (1) and subroutine TEST is called for the computation of $(\bar{x}, \bar{y})$ and the comparison of z_{min} with $z(\bar{x}, \bar{y})$ (see section 4.3 of OR [1976] for the definitions of these terms). If the test is positive (INDIC = .T.), then subroutine UPDATE is called to make the temporary change permanent. After each permanent change the solution obtained is plotted.

SUBROUTINE ALTER2 (N, NBANK, SCAL)

This subroutine corresponds to step 9 of algorithm 1 for the MTSP. The nodes that are to be temporarily placed in a different region are stored in LIST (1) to LIST (3) and subroutine TEST is called for the computation of $(\bar{x}, \bar{y})$ and the comparison of z_{min} with $z(\bar{x}, \bar{y})$. If the test is positive (INDIC = .T.), then subroutine UPDATE is called to make the temporary change permanent. After each permanent change, the solution obtained is plotted. The third column of IIHOSP is used in this subroutine to determine whether the temporary exchange under consideration has been tested before or not. If it is, a duplicate test is avoided.

SUBROUTINE TEST (I, J, N, COST, LIST, NBR, KX, INDIC)

In this subroutine the NBR hospitals given by LIST (1) to LIST (NBR) are deleted from region J and added to region I. The change in systems costs and expected emergency delivery costs are determined. New traveling salesman routes for the regions I and J are found (by calling subroutines CHULL1, BOUND1 and REFINE1) and their costs COST (I) and COST (J) are computed. The change in periodic delivery costs is determined by comparing COST (I) + COST (J) with TSCOST (I) + TSCOST (J). Then the total change in costs is determined. If there is a decrease in total costs, the test is successful (INDIC = .T.). If the test is not successful, the NBR hospitals given by LIST (1) to LIST (NBR) are reassigned to region J.

SUBROUTINE UPDATE (I, J, N, NBANK, COST, LIST, NBR, KX)

The function of this subroutine is to make the temporary change that led to a successful test (caused a decrease in total costs) permanent. The vector NUM and column 4 + KX of IHOSP have already been updated in subroutine TEST. In this subroutine IHULLA, TSCOST (I), TSCOST (J), EMCOST (I), EMCOST (J), NBLAD (I), NBLAD (J) and IIHOSP are updated.

SUBROUTINE PRINTS (N, NBANK, KX, NX, ISTOP, IUNIT)

The function of this subroutine is to provide an output in the format of the tables presented in section 4.4 of OR [1976]. ISTOP and IUNIT contain the constraints on the vehicles (maximum number of stops per vehicle and maximum number of units per vehicle) if a MVDP solution is being provided. KX selects the column of IHULLA that is to be printed and the appropriate title for the hospital assignments. "NX" is a 0, 1 variable showing whether the solution obtained is a MTSP solution or a MVDP solution. At the time this subroutine is entered the information contained in NUM (50), IBLADTR (50) and IBTR (5) should correspond to the routing that is to be printed (i.e., IBTR (1) contains the number of vehicles in region 1; NUM (1) to NUM (IBTR (1)) contain the number of stops for the vehicles in region 1; IBLADTR (1) to IBLADTR (IBTR (1)) contain the number of units carried by vehicles in region 1; IHULLA (1, 1 + KX) to IHULLA (NUM (IBTR (1)), KX + 1) contain the circuit corresponding to the first vehicle in region 1; and so on).

SUBROUTINE PLOTING (SCAL, KX, NBANK, NTRUCK)

The function of this subroutine is to plot (using the CALCOMP plotting package) the solution obtained. Subroutine PLOTING uses the information contained in either column of IHULL (150, 2) for plotting. "KX" is a variable allowing the user to plot either the first or the second column of IHULLA, and to select the appropriate title for the graph. "SCAL" contains scaling information; it should be 70 for the wide plotter and 220 for the small plotter. Arrays w, z contain the coordinates for the Chicago shoreline. The number of active vehicles in the routing to be plotted is stored in NTRUCK. At the time subroutine PLOTING is entered, the information contained in NUM (1) to NUM (NTRUCK) should correspond to the routing that is to be plotted (i.e., IHULLA (1, 1 + KX) to IHULLA (NUM (1), 2 + KX) contains the circuit corresponding to the first vehicle; IHULLA (NUM (1) + 1, 1 + KX) to IHULLA (NUM (1) + NUM (2), 1 + KX) contains the circuit corresponding to the second vehicle; and so on).

INTEGER FUNCTION DATX (I, J)

This function computes the address of the entry (I, J) of the distance matrix in the extended core storage.

References

OR, I., "Traveling Salesman Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking," Ph.D. dissertation, Northwestern University, 1976.

Appendix

The following pages contain the actual Fortran codings of the subroutines in the library BTAP and of the main program BTP.

```
      PROGRAM BTP(INPUT,OUTPUT,TAPE5,TAPE6,TAPE99,KEYBRD,CONSOL,
     1TAPE7,TAPE1=KEYBRD,TAPE2=CONSOL)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)
      DIMENSION PARM(4),KTYPE(4),ITYPE(5),CONST(2),JTYPE(2)
      LOGICAL PARM,VOP,TIMES
      DATA JTYPE/1,1/
      DATA KTYPE/3,3,3,3/
      DATA ITYPE/2,2,2,2,2/
  300 FORMAT(2X,*NUMBER OF BANKS - AT MOST 5*)
  301 FORMAT(2X,*IDENTIFICATION OF BANKS*/2X,*PROVIDE AS MANY NUMBERS
     1AS NUMBER OF BANKS*)
  302 FORMAT(//5X,*ALLOC1 IS ALLOCATION BASED ON EMERGENCY COSTS*/
     15X,*ALLOC2 IS ALLOCATION BASED ON ROUTING COSTS */
     22X,*SAY TRUE OR FALSE*/
     32X,* PLOT FOR ALLOC1, PLOT FOR ALLOC2,  ALLOC1,  ALLOC2*)
  303 FORMAT(2X,*SCALE OF PLOT*/5X,
     1*220 FOR SMALL PLOTTER, 70 FOR LARGE PLOTTER*)
  304 FORMAT(//5X,*ALTER1 IMPROVES ALLOCATIONS BY TESTING CANDIDATES 1
     .1AT A TIME*/5X,*ALTER2 IMPROVES ALLOCATIONS BY TESTING CANDIDATES
     2UP TO 3 AT A TIME*//5X,*CHOOSE ONE OR NEITHER OF THE FOLLOWING*
     3//2X,*CONTINUE WITH ALTER1,  CONTINUE WITH ALTER2*/)
  305 FORMAT(/5X,*ARE THERE CONSTRAINTS ON THE VEHICLES, IF SO WRITE*/
     15X,* T  AND A MULTI VEHICLE SOLUTION WILL BE PROVIDED*//)
  306 FORMAT(5X,*GIVE MAXIMUM NUMBER OF STOPS AND VEHICLE CAPACITY*//)
  310 FORMAT(/2X,*ARE INTERMEDIATE EXECUTION TIMES DESIRED*/
     12X,*IF SO WRITE T AND THEY WILL BE PROVIDED*//)
  311 FORMAT(/2X,*COMPUTATIONS ARE NOW STARTING, TIME IS *,F8.3/)
  312 FORMAT(/2X,*ROUTINGS BASED ON ALLOC1 ARE COMPUTED, TIME IS *,F8.3)
  313 FORMAT(/2X,*ROUTINGS BASED ON ALLOC2 ARE COMPUTED, TIME IS *,F8.3)
  314 FORMAT(/2X*ALL INDEPENDENT EXCHANGES ARE TESTED, TIME IS *,F8.3)
  315 FORMAT(2X/*DEPENDENT EXCHANGES ARE TESTED, TIME IS *,F8.3)
  316 FORMAT(2X/*MULTIPLE VEHICLE SOLUTION IS COMPUTED, TIME IS *,F8.3)
      WRITE(2,310)
      CALL FREFRM(1,TIMES,3,1,2)
      WRITE(2,300)
      CALL FREFRM(1,NBANK,2,1,2)
      WRITE(2,301)
      CALL FREFRM(NBANK,IBANK,ITYPE,1,2)
      WRITE(2,302)
      CALL FREFRM(4,PARM,KTYPE,1,2)
      WRITE(2,303)
      CALL FREFRM(1,SCAL,1,1,2)
      CALL READ1(N)
      CALL DISMAT1(N)
      X=SECOND(X)
      IF (TIMES) WRITE(2,311) X
      IF (.NOT. PARM(3)) GO TO 20
      CALL ALLOC1(N,NBANK)
      CALL TRAVEL(N,NBANK,0)
      IF (PARM(1)) CALL FLOTING(SCAL, 0,NBANK,NBANK)
      X=SECOND(X)
      IF (TIMES) WRITE(2,312) X
   20 IF (.NOT. PARM(4)) GO TO 40
      CALL ALLOC2(N,NBANK)
      CALL TRAVEL(N,NBANK,1)
```

```
PROGRAM      ETP        NORMAL                    CDC 6600 FTN V3.0-P336 OPT=1  05/25/76  10.59.07.      PAGE       2
               IF (PARM(2)) CALL PLOTING(SCAL,1,NBANK,NBANK)
               X=SECOND(X)
               IF (TIMES) WRITE(2,313) X
            40 PARM(1)=PARM(2)=.F.
               KX=0
               IF ((.NOT. PARM(3)) .AND. (.NOT. PARM(4))) KX=-2
               IF ((.NOT. PARM(3)) .OR. (.NOT. PARM(4))) GO TO 41
               WRITE (2,304)
               CALL FREFRM(2,PARM,KTYPE,1,2)
               IF (PARM(1)) CALL IMPROVE(N,NBANK,SCAL,.T.)
               IF (PARM(2)) CALL IMPROVE(N,NBANK,SCAL,.F.)
               X=SECOND(X)
               IF (PARM(1) .AND. TIMES) WRITE(2,314) X
               IF (PARM(2) .AND. TIMES) WRITE(2,315) X
            41 WRITE(2,305)
               CALL FREFRM(1,VDP,3,1,2)
               IF (.NOT. VDP) STOP
               WRITE(2,306)
               CALL FREFRM(2,CONST,JTYPE,1,2)
               IF (PARM(4)) KX=1
               IF (PARM(1) .OR. PARM(2)) KX=-1
               CALL DISPACH(N,NBANK,KX,CONST(1),CONST(2),SCAL)
               X=SECOND(X)
               IF (TIMES) WRITE(2,316) X
               END
```

```
      SUBROUTINE READ1(N)
      COMMON /E/X(150),Y(150),ALFA(150),IBLAD(150)
      COMMCN/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
  112 FORMAT(2X,10I5)
  102 FORMAT(3F5.0)
  104 FORMAT(I4,I6)
      N=1
      DO 2 I=1,150
    2 IBLAD(I)=0
    1 READ (5,102) X(N),Y(N),ALFA(N)
      IF (EOF(5))9,4
    4 CONTINUE
      N=N+1
      GO TO 1
    9 N=N-1
      DO 10 J=1,N
      READ(5,104) I,IBLD
      IBLAD(I)=IBLD
   10 CONTINUE
      READ(5,112) (SYCOST(I),I=1,120)
      RETURN
      END
```

```
      SUBROUTINE DISMAT1(N)
      COMMON /E/X(150),Y(150),ALFA(150),IBLAD(150)
      INTEGER DATX
      COMMON/CM/DMT1(150),DMT2(150)
      DO 16 I=1,N
      DO 15 J=1,N
      DUM=(Y(I)-Y(J))**2+(X(I)-X(J))**2
      DMT1(J)=SQRT(DUM)
   15 CONTINUE
      CALL WRITEC(DMT1,DATX(1,I),N)
   16 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ALLOC1(N,NBANK)
      COMMON/A/IHOSP(150,5)/E/X(150),Y(150),ALFA(150),IBLAD(150)
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
      INTEGER DATX
      COMMON/CM/DMT1(150),DMT2(150)/C/IBANK(5),NUM(50),IHULL(300,3)
  100 FORMAT(5X,10I5)
      DO 1 I=1,NBANK
      EMCOST(I)=0
      NBLAD(I)=0
      NUM(I)=0
    1 CONTINUE
      DO 5 I=1,N
      II=IBANK(1)
      CALL READEC(DMT1,DATX(1,I),N)
      DUM=DMT1(II)
      NBR=1
      DO 4 J=2,NBANK
      K=IBANK(J)
      IF (DMT1(K) .GE. DUM) GO TO 4
      DUM=DMT1(K)
      NBR=J
    4 CONTINUE
      IHOSP(I,4)=NBR
      NUM(NBR)=NUM(NBR)+1
      EMCOST(NBR)=EMCOST(NBR)+ALFA(I)*DUM
      NBLAD(NBR)=NBLAD(NBR)+IBLAD(I)
    5 CONTINUE
      WRITE(7,100) (IHOSP(I,4),I=1,N)
      RETURN
      END
```

```
      SUBROUTINE ALLOC2(N,NBANK)
      COMMON/A/IHOSP(150,5)/D/CANDID(600,4),IADRES(600)                 3300
      COMMON/CM/DMT1(150),DMT2(150)/C/IBANK(5),NUM(50),IHULL(300,3)     3310
      COMMON /E/X(150),Y(150),ALFA(150),IBLAD(150)                      3320
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)                 3330
      INTEGER DATX                                                      3340
  100 FORMAT(5X,10I5)                                                   3350
      NSIZE=KPOINT=0                                                    3360
      DO 5 I=1,N                                                        3370
    5 IHOSP(I,3)=0                                                      3380
      DO 6 J=1,NBANK                                                    3390
      EMCOST(J)=0                                                       3400
      NBLAD(J)=0                                                        3410
      JJ=IBANK(J)                                                       3420
    6 IHOSP(JJ,3)=J                                                     3430
      DO 10 I=1,NBANK                                                   3440
      II=IBANK(I)                                                       3450
      ISTAR=II                                                          3460
      CALL READEC(DMT1,DATX(1,II),N)                                    3470
      DO 7 MM=1,3                                                       3480
      DMIN=9999                                                         3490
      DO 9 J=1,N                                                        3500
      IF (IHOSP(J,3) .NE. 0) GO TO 9                                    3510
      IF (DMT1(J) .GE. DMIN) GO TO 9                                    3520
      DMIN=DMT1(J)                                                      3530
      IEND=J                                                            3540
    9 CONTINUE                                                          3550
      IHOSP(IEND,3)=I                                                   3560
      CALL AFIND(N,NSIZE,ISTAR,IEND,0)                                  3570
      ISTAR=IEND                                                        3580
    7 CONTINUE                                                          3590
      NUM(I)=4                                                          3600
      CALL AFIND(N,NSIZE,ISTAR,II,0)                                    3610
   10 CONTINUE                                                          3620
      MM=N-4*NBANK                                                      3630
      DO 30 JJ=1,MM                                                     3640
   11 KPOINT=KPOINT+1                                                   3650
      ISTAR=IADRES(KPOINT)                                              3660
      IBAR=CANDID(ISTAR,2)                                              3670
      K1=CANDID(ISTAR,3)                                                3680
      K2=CANDID(ISTAR,4)                                                3690
      IF (IHOSP(IBAR,3) .EQ. 0) GO TO 25                                3700
      CALL AFIND(N,NSIZE,K1,K2,KPOINT)                                  3710
      GO TO 11                                                          3720
   25 IGRUP=IHOSP(K1,3)                                                 3730
      IHOSP(IBAR,3)=IGRUP                                               3740
      NUM(IGRUP)=NUM(IGRUP)+1                                           3750
      CALL AFIND(N,NSIZE,K1,IBAR,KPOINT)                                3760
      CALL AFIND(N,NSIZE,IBAR,K2,KPOINT)                                3770
   30 CONTINUE                                                          3780
      DO 40 I=1,N                                                       3790
      IHOSP(I,5)=IHOSP(I,3)                                             3800
      IB=IHOSP(I,3)                                                     3810
      IBK=IBANK(IB)                                                     3820
      CALL READEC(DMTX,DATX(I,IBK),1)                                   3830
```

```
      EMCOST(IB)=EMCOST(IB)+ALFA(I)*DMTX
   39 NBLAD(IB)=NBLAD(IB)+IBLAD(I)
   40 CONTINUE
      WRITE(7,100) (IHOSP(I,5),I=1,N)
      RETURN
      END
```

SUBROUTINE TRAVEL NORMAL CDC 6600 FTN V3.0-P336 OPT=1 05/20/76 14.00.10. PAGE 1

```
      SUBROUTINE TRAVEL(N,NBANK,KX)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)/B/IHULLA(150,2)
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
      ISTAR=0
      DO 5 II=1,NBANK
      MEM=NUM(II)
      CALL CHULL1(N,IC,KX,II)
      CALL BOUND1(N,IC,TSCOST,KX,II)
      IF (NUM(II) .GE. 5) CALL REFINE1(N,TSCOST,KX,II)
      DO 6 I=1,MEM
    6 IHULLA(ISTAR+I,KX+1)=IHULL(I,1)
      ISTAR=ISTAR+NUM(II)
    5 CONTINUE
      CALL PRINTS(N,NBANK,KX,1)
      RETURN
      END
```

```
      SUBROUTINE DISPACH(N,NBANK,KX,STOPS,UNITS,SCAL)
      COMMON/CM/DMT1(150),DMT2(150)/A/IHOSP(150,5)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)/B/IHULLA(150,2)
      COMMON/F/IBLADTR(50),IBTR(5),R(150),TETA(150),IPOL(100),NUMTR(50)
     1,COST(9),IBLADT(9),IDUM(352)
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
      COMMON/E/X(150),Y(150),ALFA(1=0),IBLAD(150)
  100 FORMAT(5X,10I5)
  200 FORMAT(1H1/9X,*THE FOLLOWING ALLOCATION IS SUPPLIED EXTERNALLY*//)
  210 FORMAT(3X,*HOSPITAL*,I4,6X,*BANK*,I3)
      KK=KX
      IF (KX .LT. 0) KK=0
      IF (KX .NE. -2) GO TO 3
      READ (7,100) (IHOSP(I,4+KK), I=1,N)
      WRITE(6,200)
      WRITE(6,210) (I,IHOSP(I,4+KK), I=1,N)
    3 DO 5 IB=1,NBANK
      INUM=IBANK(IB)
      EMCOST(IB)=NBLAD(IB)=0
      CALL READEC(DMT1,DATX(1,INUM),N)
      DO 4 I=1,N
      IF (IHOSP(I,4+KK) .NE. IB) GO TO 4
      EMCOST(IB)=EMCOST(IB)+DMT1(I)
      NBLAD(IB)=NBLAD(IB)+IBLAD(I)
      IF (INUM .EQ. I) GO TO 4
      R(I)=DMT1(I)
      IF (X(I) .EQ. X(INUM)) GO TO 1
      TETA(I)=ATAN((Y(I)-Y(INUM))/(X(I)-X(INUM)))
      IF ((Y(I)-Y(INUM))/(X(I)-X(INUM)) .LT. 0) TETA(I)=TETA(I)+3.141593
      IF (Y(I) .GT. Y(INUM)) GO TO 4
      IF (Y(I) .EQ. Y(INUM)) GO TO 2
      TETA(I)=TETA(I)+3.141593
      GO TO 4
    1 IF (Y(I) .GE. Y(INUM)) TETA(I)=3.141593/2
      IF (Y(I) .LT. Y(INUM)) TETA(I)=(3*3.141593)/2
      GO TO 4
    2 IF (X(I) .LT. X(INUM)) TETA(I)=3.141593
    4 CONTINUE
    5 CONTINUE
      ISTAR=JSTAR=0
      DO 50 IB=1,NBANK
      INUM=IBANK(IB)
      TSCOST(IB)=IBTR(IB)=NS=0
      DO 15 J=1,N
      IHOSP(J,5-KK)=0
      IF (IHOSP(J,4+KK) .NE. IB) GO TO 15
      IF (INUM .EQ. J) GO TO 15
      NS=NS+1
      IF (NS .EQ. 1) GO TO 10
      IEND=NS-1
      DO 7 I=1,IEND
      JPOINT=IPOL(I)
      IF (TETA(J) .LT. TETA(JPOINT)) GO TO 8
      IF (TETA(J) .GT. TETA(JPOINT)) GO TO 7
      IF (R(J) .LT. R(JPOINT)) GO TO 8
```

```
    7 CONTINUE                                                          8640
   10 IPOL(NS)=J                                                        8650
      GO TO 15                                                          8660
    8 JEND=NS-I                                                         8670
      DO 9 JJ=1,JEND                                                    8680
    9 IPOL(NS+1-JJ)=IPOL(NS-JJ)                                         8690
      IPOL(I)=J                                                         8700
   15 CONTINUE                                                          8710
      WRITE(6,100) (IPOL(J),J=1,NS)                                     8720
      IBLD=NM=0                                                         8730
      NTR=1                                                             8740
      DO 30 J=1,NS                                                      8750
      I=IPOL(J)                                                         8760
      BPERDAY=IBLAD(I)/260.0                                            8770
      IF ((IBLD+BPERDAY .GT. UNITS) .OR. (NM+1 .GT. STOPS)) GO TO 25    8780
      IBLD=IBLD+BPERDAY + 0.999                                         8790
      NM=NM+1                                                           8800
      IHOSP(I,5-KK)=NTR                                                 8810
      GO TO 30                                                          8820
   25 NUM(NTR)=NM                                                       8830
      IBLADT(NTR)=IBLD                                                  8840
      NTR=NTR+1                                                         8850
      NM=1                                                              8860
      IBLD=BPERDAY+0.999                                                8870
      IHOSP(I,5-KK)=NTR                                                 8880
   30 CONTINUE                                                          8890
      NUM(NTR)=NM                                                       8900
      IBLADT(NTR)=IBLD                                                  8910
      IBTR(IB)=NTR                                                      8920
      WRITE(6,100) NTR,(NUM(J),J=1,NTR)                                 8930
      DO 40 J=1,NTR                                                     8940
      IHOSP(INUM,5-KK)=J                                                8950
      NUM(J)=NUM(J)+1                                                   8960
      CALL CHULL1(N,IC,1-KK,J)                                          8970
      CALL BOUND1(N,IC,COST,1-KK,J)                                     8980
      IF (NUM(J) .GE. 6) CALL REFINE1(N,COST,1-KK,J)                    8990
      NS=NUM(J)                                                         9000
      DO 41 I=1,NS                                                      9010
   41 IHULLA(ISTAR+I,KK+1)=IHULL(I,1)                                   9020
      TSCOST(IB)=TSCOST(IB)+COST(J)                                     9030
      NUMTR(JSTAR+J)=NUM(J)                                             9040
      IBLADTR(JSTAR+J)=IBLADT(J)                                        9050
      ISTAR=ISTAR+NUM(J)                                                9060
   40 CONTINUE                                                          9070
      JSTAR=JSTAR+NTR                                                   9080
   50 CONTINUE                                                          9090
      DO 55 J=1,JSTAR                                                   9100
   55 NUM(J)=NUMTR(J)                                                   9110
      I=STOPS                                                           9111
      J=UNITS                                                           9112
      CALL PRINTS(N,NBANK,KX,2,I,J)                                     9120
      CALL PLOTING(SCAL,KX,NBANK,JSTAR)                                 9130
      END                                                               9140
```

```
      SUBROUTINE CHULL1(N,IC,KX,IX)
      COMMON/A/IHCSP(150,5)/E/X(150),Y(150),ALFA(150),IBLAD(150)        830
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)                            840
      IC=0                                                              850
      DO 1 I=1,N                                                        860
      IF (IHOSP(I,4+KX) .EQ. IX) GO TO 2                                870
    1 CONTINUE                                                          880
      STOP                                                              890
    2 BEST=X(I)                                                         900
      MEND=I                                                            910
      XLAST=X(I)                                                        920
      MSTAR=I                                                           930
      DO 5 I=1,N                                                        940
    5 IHOSP(I,3)=0                                                      950
      ISTART=I+1                                                        960
      DO 10 J=ISTART,N                                                  970
      IF (IHOSP(J,4+KX) .NE. IX) GO TO 10                               980
      IF (X(J) .GE. BEST) GO TO 8                                       990
      MSTAR=J                                                          1000
      BEST=X(J)                                                        1010
    8 IF (X(J) .LE. XLAST) GO TO 10                                    1020
      MEND=J                                                           1030
      XLAST=X(J)                                                       1040
   10 CONTINUE                                                         1050
      IHOSP(MSTAR,3)=1                                                 1060
      M=MSTAR                                                          1070
   11 BEST=-10000                                                      1080
      DO 20 I=1,N                                                      1090
      IF (IHOSP(I,4+KX) .NE. IX) GO TO 20                              1100
      IF (X(I)-X(M)) 20,16,17                                          1110
   16 IF (Y(I) .EQ. Y(M)) GO TO 20
      IF (Y(I) .GT. Y(M)) SLOPE=9999
      IF (Y(I) .LT. Y(M)) SLOPE=-9999                                  1112
      GO TO 18                                                         1113
   17 SLOPE=(Y(I)-Y(M))/(X(I)-X(M))                                    1120
   18 IF (SLOPE .LE. BEST) GO TO 20                                    1130
      BEST=SLOPE                                                       1140
      MNEXT=I                                                          1150
   20 CONTINUE                                                         1160
      IHOSP(MNEXT,3)=IC+2                                              1170
      CALL PFIND(N,IC,M,MNEXT,0,KX,IX)                                 1180
      IHULL(IC,1)=M                                                    1190
      IHULL(IC,2)=MNEXT                                                1200
      M=MNEXT                                                          1210
      IF (M .NE. MEND) GO TO 11                                        1220
   21 BEST=-10000                                                      1230
      DO 30 I=1,N                                                      1240
      IF (IHOSP(I,4+KX) .NE. IX) GO TO 30                              1250
      IF (X(I)-X(M)) 24,23,30                                          1260
   23 IF (Y(I) .EQ. Y(M)) GO TO 30
      IF (Y(I) .LT. Y(M)) SLOPE=9999
      IF (Y(I) .GT. Y(M)) SLOPE=-9999                                  1262
      GO TO 25                                                         1263
   24 SLOPE=(Y(I)-Y(M))/(X(I)-X(M))                                    1270
   25 IF (SLOPE .LE. BEST) GO TO 30                                    1280
```

```
      BEST=SLOPE
      MNEXT=I
   30 CONTINUE
      IHOSP(MNEXT,3)=IC+2
      CALL PFIND(N,IC,M,MNEXT,0,KX,IX)
      IHULL(IC,1)=M
      IHULL(IC,2)=MNEXT
      M=MNEXT
      IF (M .NF. MSTAR) GO TO 21
      RETURN
      END
```

```
      SUBROUTINE PFIND(N,NSIZE,K1,K2,KPOINT,KX,IX)
      COMMON/A/IHOSP(150,5)/D/CANDID(600,4),IADRES(600)
      COMMON /E/X(150),Y(150),ALFA(150),IBLAD(150)
      INTEGER DATX
      COMMON /CM/DMT1(150),DMT2(150)
      BESS=100000
      CALL READEC(DMT2,DATX(1,K2),N)
      CALL READEC(DMT1,DATX(1,K1),N)
   19 DO 20 I=1,N
      IF (IHOSP(I,4+KX) .NE. IX) GO TO 20
      IF (IHOSP(I,3) .GE. 1) GO TO 20
      DIF=DMT1(I)+DMT2(I)-DMT1(K2)
      ANG=(DMT1( I)+DMT2(I))/DMT1( K2)
      DIS=DIF*ANG
      IF (DIS .GT. BESS) GO TO 20
      BESS=DIS
      IBEST=I
   20 CONTINUE
      GO TO 25
      ENTRY AFIND
      BESS=100000
      CALL READEC(DMT2,DATX(1,K2),N)
      CALL READEC(DMT1,DATX(1,K1),N)
      YMID=(Y(K1)+Y(K2))/2
      XMID=(X(K1)+X(K2))/2
      DO 10 I=1,N
      IF (IHOSP(I,3) .GE. 1) GO TO 10
      DMID=SQRT((XMID-X(I))**2+(YMID-Y(I))**2)
      DIS=AMIN1(DMT1(I),DMT2(I),DMID)
      COSI=(DMT1(I)**2+DMT2(I)**2-DMT1(K2)**2)/
     1(2*DMT1(I)*DMT2(I))
      DIS=DIS+COSI
      IF (DIS .GT. BESS) GO TO 10
      BESS=DIS
      IBEST=I
   10 CONTINUE
   25 NSIZE=NSIZE+1
      CANDID(NSIZE,1)=BESS
      CANDID(NSIZE,2)=IBEST
      CANDID(NSIZE,3)=K1
      CANDID(NSIZE,4)=K2
      CALL ADRES(NSIZE,KPOINT)
      RETURN
      END
```

```
      SUBROUTINE ADRES(NSIZE,KPOINT)
      COMMON/A/IHOSP(150,5)/D/CANDID(600,4),IADRES(600)
      IF (NSIZE .EQ. 1) GO TO 10
      ISTAR=KPOINT+1
      IEND=NSIZE-1
      DO 1 I=ISTAR,IEND
      JPOINT=IADRES(I)
      IF (CANDID(JPOINT,1) .GT. CANDID(NSIZE,1)) GO TO 8
    1 CONTINUE
   10 IADRES(NSIZE)=NSIZE
      RETURN
    8 JEND=NSIZE-I
      DO 9 J=1,JEND
    9 IADRES(NSIZE+1-J)=IADRES(NSIZE-J)
      IADRES(I)=NSIZE
      RETURN
      END
```

```
      SUBROUTINE BOUND1(N,IC,COST,KX,IX)
      COMMON/A/IHOSP(150,5)/D/CANDID(600,4),IADRES(600)
      INTEGER DATX
      COMMON/CM/DMT1(150),DMT2(150)/C/IBANK(5),NUM(50),IHULL(300,3)
      DIMENSION COST(9)
      KPOINT=0
      NSIZE=IC
      ICOUNT=IC
      DO 1 I=1,300
    1 IHULL(I,3)=0
      MM=NUM(IX)-IC
      IF (MM .EQ. 0) GO TO 31
      DO 30 JJ=1,MM
   11 KPOINT=KPOINT+1
      ISTAR=IADRES(KPOINT)
      IBAR=CANDID(ISTAR,2)
      K1=CANDID(ISTAR,3)
      K2=CANDID(ISTAR,4)
      IF (IHOSP(IBAR,3) .EQ. 0) GO TO 25
      CALL PFIND(N,NSIZE,K1,K2,KPOINT,KX,IX)
      GO TO 11
   25 DO 26 I=1,IC
      IF (IHULL(I,1) .EQ. K1) IHULL(I,3)=1
   26 CONTINUE
      ICOUNT=ICOUNT+1
      IHOSP(IBAR,3)=ICOUNT
      IC =IC+1
      IHULL(IC,1)=K1
      IHULL(IC,2)=IBAR
      CALL PFIND(N,NSIZE,K1,IBAR,KPOINT,KX,IX)
      IC=IC+1
      IHULL(IC,1)=IBAR
      IHULL(IC,2)=K2
      CALL PFIND(N,NSIZE,IBAR,K2,KPOINT,KX,IX)
   30 CONTINUE
   31 COST(IX)=0
      DO 40 I=1,IC
      IF (IHULL(I,3) .EQ. 1) GO TO 40
      ISTAR=IHULL(I,1)
      IHOSP(ISTAR,2)=IHULL(I,2)
      IEND=IHULL(I,2)
      IHOSP(IEND,1)=IHULL(I,1)
      CALL READEC(DMTX,DATX(ISTAR,IEND),1)
      COST(IX)=COST(IX)+DMTX
   40 CONTINUE
      DO 41 II=1,N
      IF (IHOSP(II,KX+4) .EQ. IX) GO TO 43
   41 CONTINUE
      STOP
   43 IPREV=II
      INEXT=IHOSP(II,2)
      IEND=NUM(IX)
      DO 45 I=1,IEND
      IHULL(I,1)=IPREV
      IHULL(I,2)=INEXT
```

```
SUBROUTINE  BOUND1     NORMAL                     CDC 6600 FTN V3.0-P336 OPT=1  05/20/76  14.00.10.      PAGE      2

              IPREV=INEXT                                              2120
              INEXT=IHOSP(INEXT,2)                                     2130
           45 CONTINUE                                                 2140
              RETURN                                                   2150
              END                                                      2160
```

```
      SUBROUTINE REFINE1(N,COST,KX,IX)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)                        2180
      INTEGER DATX                                                  2190
      COMMON/CM/DMT1(150),DMT2(150)/A/IHOSP(150,5)                  2200
      DIMENSION COST(9)                                             2210
      DO 11 JJ=1,2                                                  2220
      KK=3-JJ                                                       2230
      DO 11 I=1,N                                                   2240
      IF (IHOSP(I,4+KX) .NE. IX) GO TO 12                           2250
      I1=I                                                          2260
      J1=I2=IHOSP(I,1)                                              2270
      DO 6 IN=1,KK                                                  2280
    6 I2=IHOSP(I2,2)                                                2290
      J2=IHOSP(I2,2)                                                2300
      CALL READEC(DMT2,DATX(1,J2),N)                                2310
      CALL READEC(DMT1,DATX(1,J1),N)                                2320
      DIF1=DMT1(I1)+DMT2(I2)-DMT1(J2)                               2330
      ISTAR=J2                                                      2340
      IEND=NUM(IX)-KK-1                                             2350
      DO 10 J=1,IEND                                                2360
      K1=ISTAR                                                      2370
      K2=IHOSP(ISTAR,2)                                             2380
      CALL READEC(DMT2,DATX(1,K2),N)                                2390
      CALL READEC(DMT1,DATX(1,K1),N)                                2400
      DIF2=DMT1(I2)+DMT2(I1)-DMT1(K2)                               2410
      IF (DIF1 .GT. DIF2) GO TO 7                                   2420
      ISTAR=IHOSP(ISTAR,2)                                          2430
   10 CONTINUE                                                      2440
      GO TO 12                                                      2450
    7 CALL CHANGE(I1,I2,J1,J2,K1,K2,N)                              2460
   12 CONTINUE                                                      2470
   11 CONTINUE                                                      2480
      COST(IX)=0                                                    2490
      DO 13 J=1,N                                                   2500
      IF (IHOSP(J,4+KX) .EQ. IX) GO TO 14                           2510
   13 CONTINUE                                                      2520
      STOP                                                          2530
   14 IPREV=J                                                       2540
      INEXT=IHOSP(J,2)                                              2550
      IEND=NUM(IX)                                                  2560
      DO 15 I=1,IEND                                                2570
      CALL READEC(DMTX,DATX(IPREV,INEXT),1)                         2580
      COST(IX)=COST(IX)+DMTX                                        2590
      IHULL(I,1)=IPREV                                              2600
      IHULL(I,2)=INEXT                                              2610
      IPREV=INEXT                                                   2620
      INEXT=IHOSP(INEXT,2)                                          2630
   15 CONTINUE                                                      2640
      RETURN                                                        2650
      END                                                           2660
```

```
      SUBROUTINE CHANGE(I1,I2,J1,J2,K1,K2,N)
      COMMON /A/ IHOSP(150,5)
      IHOSP(J1,2)=J2
      IHOSP(J2,1)=J1
      IHOSP(K1,2)=I2
      IHOSP(K2,1)=I1
      IHOSP(I1,1)=K2
      INEXT=I2
      IPREV=K1
   10 IHOSP(INEXT,2)=IHOSP(INEXT,1)
      IHOSP(INEXT,1)=IPREV
      IF (INEXT .EQ. I1) GO TO 15
      IPREV=INEXT
      INEXT=IHOSP(INEXT,2)
      GO TO 10
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE IMPROVE(N,NBANK,SCAL,EXINDP)
      INTEGER DATX
      LOGICAL EXINDP
      COMMON/F/IALTER(5,5,10),NALTER(5,5),IIHOSP(150,3),IDUM(150)
      COMMON/CM/DMT1(150),DMT2(150)/A/IHOSP(150,5)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)/B/IHULLA(150,2)
  110 FORMAT(1H1//5X,*LIST OF POSSIBLE EXCHANGES*/)
  120 FORMAT(///5X,*HOSPITALS IN GROUP*,I3,* TO BE TRIED IN GROUP*
     1,I3/)
  121 FORMAT(2X,10I5)
      DO 1 I=1,NBANK
      DO 1 J=1,NBANK
      NALTER(I,J)=0
      DO 1 K=1,10
      IALTER(I,J,K)=0
    1 CONTINUE
      DO 10 I=1,N
      IIHOSP(I,3)=-1
      IF (IHOSP(I,4) .EQ. IHOSP(I,5)) GO TO 10
      II=IHOSP(I,4)
      JJ=IHOSP(I,5)
      IB=IBANK(JJ)
      CALL READEC(DMT2,DATX(1,IB),N)
      IB=IBANK(II)
      CALL READEC(DMT1,DATX(1,IB),N)
      NALTER(II,JJ)=NALTER(II,JJ)+1
      IF (NALTER(II,JJ) .GT. 10) NALTER(II,JJ)=10
      NM=NALTER(II,JJ)
      JEND=NM-1
      IF (NM .EQ. 1) GO TO 7
       DO 6 J=1,JEND
      JX=IALTER(II,JJ,J)
      IF (DMT1(I)-DMT2(I) .LE. DMT1(JX)-DMT2(JX)) GO TO 8
    6 CONTINUE
      IF (NM .LT. 10) IALTER(II,JJ,NM)=I
      JX=IALTER(II,JJ,NM)
      IF (DMT1(I)-DMT2(I) .LE. DMT1(JX)-DMT2(JX)) IALTER(II,JJ,NM)=I
      GO TO 10
    8 JEND=NM-J
      DO 9 J=1,JEND
      IALTER(II,JJ,NM+1-J)=IALTER(II,JJ,NM-J)
    9 CONTINUE
    7 IALTER(II,JJ,NM-JEND)=I
   10 CONTINUE
      WRITE(6,110)
      DO 12 I=1,NBANK
      DO 11 J=1,NBANK
      IF (NALTER(I,J) .EQ. 0) GO TO 11
      WRITE(6,120) J,I
      NM=NALTER(I,J)
      WRITE(6,121) (IALTER(I,J,K),K=1,NM)
   11 CONTINUE
   12 CONTINUE
      DO 15 I=1,N
      IHOSP(I,4)=IHOSP(I,5)
```

```
      IIHOSP(I,1)=IHOSP(I,1)                                          7740
      IIHOSP(I,2)=IHOSP(I,2)                                          7750
      IHULLA(I,1)=IHULLA(I,2)                                         7760
   15 CONTINUE                                                        7770
      IF (EXINDP) CALL ALTER1(N,NBANK,SCAL)                           7780
      IF (.NOT. EXINDP) CALL ALTER2(N,NBANK,SCAL)                     7790
      CALL PRINTS(N,NBANK,-1,1)                                       7800
      RETURN                                                          7810
      END                                                             7820
```

```
      SUBROUTINE ALTER1(N,NBANK,SCAL)
      LOGICAL INDIC
      COMMON/F/IALTER(5,5,10),NALTER(5,5),IIHOSP(150,3),IDUM(150)
      DIMENSION COST(9),LIST(5)
      IEND=NBANK-1
      DO 30 I=1,IEND
      JSTAR=I+1
      DO 29 J=JSTAR,NBANK
      ITER=MAX0(NALTER(I,J),NALTER(J,I))
      IF (ITER .EQ. 0) GO TO 29
      DO 25 K1=1,ITER
      IF (K1 .GT. NALTER(I,J)) GO TO 20
      LIST(1)=IALTER(I,J,K1)
      CALL TEST(I,J,N,COST,LIST,1,1,INDIC)
      IF (INDIC) CALL UPDATE(I,J,N,NBANK,COST,LIST,1,1)
      IF (INDIC) CALL PLOTING(SCAL,-1,NBANK,NBANK)
   20 IF (K1 .GT. NALTER(J,I)) GO TO 25
      LIST(1)=IALTER(J,I,K1)
      CALL TEST(J,I,N,COST,LIST,1,0,INDIC)
      IF (INDIC) CALL UPDATE(J,I,N,NBANK,COST,LIST,1,0)
      IF (INDIC) CALL PLOTING(SCAL,-1,NBANK,NBANK)
   25 CONTINUE
   29 CONTINUE
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ALTER2(N,NBANK,SCAL)
      LOGICAL INDIC
      COMMON/F/IALTER(5,5,10),NALTER(5,5),IIHOSP(150,3),IDUM(150)
      DIMENSION COST(9),LIST(5)
      IEND=NBANK-1
      DO 30 I=1,IEND
      JSTAR=I+1
      DO 29 J=JSTAR,NBANK
      ITER=MAX0(NALTER(I,J),NALTER(J,I))
      IF (ITER .EQ. 0) GO TO 29
      DO 25 K1=1,ITER
      IZ=I
      JZ=J
      KX=1
   15 IF (K1 .GT. NALTER(IZ,JZ)) GO TO 20
      J1=IALTER(IZ,JZ,K1)
      IF (IIHOSP(J1,3) .EQ. 0) GO TO 20
      NBR=1
      LIST(1)=J1
      CALL TEST(IZ,JZ,N,COST,LIST,NBR,KX,INDIC)
      IF (INDIC) GO TO 19
      J2=IIHOSP(J1,1)
      IF (IIHOSP(J1,3) .EQ. J2) GO TO 16
      IIHOSP(J1,3)=J2
      LIST(2)=J2
      NBR=2
      CALL TEST(IZ,JZ,N,COST,LIST,NBR,KX,INDIC)
      IF (INDIC) GO TO 19
   16 J3=IIHOSP(J1,2)
      IF (IIHOSP(J3,3) .EQ. J1) GO TO 17
      IIHOSP(J3,3)=J1
      NBR=2
      LIST(2)=J3
      CALL TEST(IZ,JZ,N,COST,LIST,NBR,KX,INDIC)
      IF (INDIC) GO TO 19
   17 LIST(2)=J2
      LIST(3)=J3
      NBR=3
      CALL TEST(IZ,JZ,N,COST,LIST,NBR,KX,INDIC)
      IF (INDIC) GO TO 19
      GO TO 20
   19 CALL UPDATE(IZ,JZ,N,NBANK,COST,LIST,NBR,KX)
      CALL PLOTING(SCAL,-1,NBANK,NBANK)
   20 IF (IZ .EQ. J) GO TO 25
      IZ=J
      JZ=I
      KX=0
      GO TO 15
   25 CONTINUE
   29 CONTINUE
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TEST(I,J,N,COST,LIST,NBR,KX,INDIC)
      INTEGER DATX                                                      6560
      LOGICAL INDIC                                                     6570
      COMMON/F/IALTER(5,5,10),NALTER(5,5),IIHOSP(150,3),IDUM(150)       6580
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)                 6590
      COMMON/CM/DMT1(150),DMT2(150)/C/IBANK(5),NUM(50),IHULL(300,3)     6600
      COMMON /E/X(150),Y(150),ALFA(150),IBLAD(150)/A/IHOSP(150,5)       6610
      DIMENSION COST(9),LIST(5)                                         6620
  100 FORMAT(1H1/10X,*TEST DATA*//)                                     6630
  110 FORMAT(///2X,*ORIGINAL ROUTING COST FOR GROUP*,I3,*=*,F9.3/       6640
     12X,*REVISED ROUTING COST FOR GROUP*,I3,*=*,F9.3//)                6650
  120 FORMAT(/2X,*MARGINAL DIFFERENCE IN EMERFENCT COST=*,F9.3///)      6660
  130 FORMAT(2X,*MARGINAL DIFFERENCE IN SYSTEM COST=*,F9.3///)          6670
  140 FORMAT(2X,*TOTAL MARGINAL DIFFERENCE=*,F9.3///)                   6680
  153 FORMAT(2X,10I5)                                                   6690
  160 FORMAT(//5X,*HOSPITALS ASSIGNED TO GROUP*,I3,                     6700
     1*  FROM GROUP*,I3,* :*,5I5//)                                     6710
  200 FORMAT(//2X,*REVISED ROUTING FOR GROUP*,I3/)                      6720
      EMDIF=0                                                           6730
      INDIC=.F.                                                         6740
      NUM(I)=NUM(I)+NBR                                                 6750
      NUM(J)=NUM(J)-NBR                                                 6760
      DO 1 II=1,NBR                                                     6770
      JJ=LIST(II)                                                       6780
    1 IHOSP(JJ,4+KX)=I                                                  6790
      CALL CHULL1(N,IC,KX,I)                                            6800
      CALL BOUND1(N,IC,COST,KX,I)                                       6810
      IF (NUM(I) .GE. 5) CALL REFINE1(N,COST,KX,I)                      6820
      NI=NUM(I)                                                         6830
      DO 5 II=1,NI                                                      6840
    5 IDUM(II)=IHULL(II,1)                                              6850
      CALL CHULL1(N,IC,KX,J)                                            6860
      CALL BOUND1(N,IC,COST,KX,J)                                       6870
      IF (NUM(J) .GE. 5) CALL REFINE1(N,COST,KX,J)                      6880
      NJ=NUM(J)                                                         6890
      TSDIF=COST(I)+COST(J)-TSCOST(I)-TSCOST(J)                         6900
      SYDIF=SYCOST(NJ)+SYCOST(NI)-SYCOST(NJ+NBR)-SYCOST(NJ-NBR)         6910
      CALL READEC(DMT1,DATX(1,I),N)                                     6920
      CALL READEC(DMT2,DATX(1,J),N)                                     6930
      DO 10 II=1,NBR                                                    6940
      I1=LIST(II)                                                       6950
      EMDIF=EMDIF+ALFA(I1)*(DMT1(I1)-DMT2(I1))                          6960
   10 CONTINUE                                                          6970
      DIF=TSDIF+EMDIF+SYDIF                                             6980
      IF (DIF .LE. 0) INDIC=.T.                                         6990
      WRITE(6,100)                                                      7000
      WRITE(6,160) I,J,(LIST(II),II=1,NBR)                              7010
      WRITE(6,200) I                                                    7020
      WRITE(6,153) (IDUM(II),II=1,NI)                                   7030
      WRITE(6,200) J                                                    7040
      WRITE(6,153) (IHULL(II,1),II=1,NJ)                                7050
      WRITE(6,110) I,TSCOST(I),I,COST(I)                                7060
      WRITE(6,110) J,TSCOST(J),J,COST(J)                                7070
      WRITE(6,120) EMDIF                                                7080
      WRITE(6,130) SYDIF                                                7090
```

```
      WRITE(6,140) DIF
      IF (INDIC) RETURN
      NUM(I)=NUM(I)-NBR
      NUM(J)=NUM(J)+NBR
      DO 15 II=1,NBR
      JJ=LIST(II)
   15 IHOSP(JJ,4+KX)=J
      RETURN
      END
```

```
      SUBROUTINE UPDATE(I,J,N,NBANK,COST,LIST,NBR,KX)
      COMMON/CM/DMT1(150),DMT2(150)/C/IBANK(5),NUM(50),IHULL(300,3)
      COMMON/F/IALTER(5,5,10),NALTER(5,5),IIHOSP(150,3),IDUM(150)
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
      COMMON/B/IHULLA(150,2)/E/X(150),Y(150),ALFA(150),IBLAD(150)
      COMMON/A/IHOSP(150,5)
      DIMENSION COST(9),LIST(5)
      ISTARX=ISTAR=0
      DO 40 II=1,NBANK
      IEND=NUM(II)
      IF (II .EQ .I) GO TO 20
      IF (II .EQ. J) GO TO 30
      DO 15 JJ=1,IEND
      IHULLA(ISTAR+JJ,1)=IHULLA(ISTARX+JJ,1)
   15 CONTINUE
      ISTARX=ISTARX+NUM(II)
      GO TO 39
   20 ISTARX=ISTARX+NUM(II)-NBR
      GO TO 39
   30 DO 35 JJ=1,IEND
      IHULLA(ISTAR+JJ,1)=IHULL(JJ,1)
   35 CONTINUE
      ISTARX=ISTARX+NUM(II)+NBR
   39 ISTAR=ISTAR+NUM(II)
   40 CONTINUE
      ISTAR=0
      DO 45 II=1,NBANK
      IF (II .NE. I) GO TO 44
      IEND=NUM(II)
      DO 41 JJ=1,IEND
      IHULLA(ISTAR+JJ,1)=IDUM(JJ)
   41 CONTINUE
      GO TO 46
   44 ISTAR=ISTAR+NUM(II)
   45 CONTINUE
   46 TSCOST(I)=COST(I)
      TSCOST(J)=COST(J)
      DO 50 II=1,NBR
      JJ=LIST(II)
      IIHOSP(JJ,3)=0
      EMCOST(I)=EMCOST(I)+ALFA(JJ)*DMT1(JJ)
      EMCOST(J)=EMCOST(J)-ALFA(JJ)*DMT2(JJ)
      NBLAD(I)=NBLAD(I)+IBLAD(JJ)
      NBLAD(J)=NBLAD(J)-IBLAD(JJ)
      IHOSP(JJ,5-KX)=I
   50 CONTINUE
      DO 51 II=1,N
      IIHOSP(II,1)=IHOSP(II,1)
      IIHOSP(II,2)=IHOSP(II,2)
   51 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINTS(N,NBANK,KX,NX,ISTOP,IUNIT)
      COMMON/C/IBANK(5),NUM(50),IHULL(300,3)/B/IHULLA(150,2)
      COMMON/K/TSCOST(5),EMCOST(5),NBLAD(5),SYCOST(120)
      COMMON/F/IBLADTR(50),IBTR(5),TDUM(820)
  110 FORMAT(// ,5X,*OPTIMAL ALLOCATION AND ROUTING */6X,*ALLOCATION BAS
     1ED ON EMERGENCY COSTS ONLY*//)
  111 FORMAT (///3X,*BANK *,I2,* , IDENTIFICATION-HOSPITAL*,I4/3X,*ROUTI
     1NG*/)
  109 FORMAT(2X,5F10.3)
  112 FORMAT(10X,10I5)
  113 FORMAT(///3X,*EMERGENCY COST*,9X,F9.2//3X,*ROUTINE DELIVARY COST *
     1,F10.2//3X,*SYSTEM COST*,12X,F9.2//3X,*TOTAL COST*,13X,F9.2//)
  120 FORMAT(// ,5X,*OPTIMAL ALLOCATION AND ROUTING */6X,*ALLOCATION BAS
     1ED ON ROUTING COSTS ONLY*//)
  125 FORMAT(//5X,*NUMBER OF HOSPITALS IN THE SYSTEM *I7/)
  126 FORMAT(/5X,*AMOUNT OF BLOOD USED IN THE SYSTEM*I7/)
  130 FORMAT(// ,5X,*OPTIMAL ALLOCATION AND ROUTING */6X,*ALLOCATION BAS
     1ED ON ROUTING,EMERGENCY AND SYSTEM COSTS*//)
  210 FORMAT(1H1/40X,*SINGLE VEHICLE SOLUTION*/)
  220 FORMAT(1H1/40X,*MULTI VEHICLE SOLUTION*//5X,*CONSTRAINTS:*,4X,
     1*MAXIMUM NUMBER OF STOPS-*,I4/21X,*MAXIMUM NUMBER OF UNITS-*,I4/)
  230 FORMAT(/6X,*TRUCK NO*,I3,6X,*NUMBER OF STOPS*,I3,6X,*NUMBER OF UNI
     1TS*I5/)
      KK=KX
      IF (KX .LT. 0) KK=0
      IF (NX .EQ. 1) WRITE(6,210)
      IF (NX .EQ. 2) WRITE(6,220) ISTOP,IUNIT
      IF (KX .EQ. 0) WRITE(6,110)
      IF (KX .EQ. 1) WRITE(6,120)
      IF (KX .EQ. -1) WRITE(6,130)
      ECOST=RCOST=SCOST=ISTAR=JSTAR=IPREV=0
      DO 6 II=1,NBANK
      WRITE(6,111) II,IBANK(II)
      IF (NX .EQ. 1) GO TO 4
      NTR=IBTR(II)
      DO 3 J=1,NTR
      WRITE(6,230) J, NUM(JSTAR+J),IBLADTR(JSTAR+J)
      JEND=NUM(JSTAR+J)
      WRITE(6,112) (IHULLA(ISTAR+I,KK+1),I=1,JEND)
      ISTAR=ISTAR+JEND
    3 CONTINUE
      MEMBER=ISTAR-IPREV-NTR+1
      IPREV=ISTAR
      WRITE(6,125) MEMBER
      JSTAR=JSTAR+NTR
      GO TO 5
    4 MEMBER=NUM(II)
      WRITE(6,112) (IHULLA(ISTAR+I,1+KK),I=1,MEMBER)
      WRITE(6,125) NUM(II)
      ISTAR=ISTAR+NUM(II)
    5 WRITE(6,126) NBLAD(II)
      ECOST=ECOST+EMCOST(II)
      RCOST=RCOST+TSCOST(II)
      SCOST=SYCOST(MEMBER)+SCOST
    6 CONTINUE
```

```
      TOT=ECOST+RCOST+SCOST
      WRITE(6,113) ECOST,RCOST,SCOST,TOT
      RETURN
      END
```

```
      SUBROUTINE PLOTING(SCAL,KX,NBANK,NTRUCK)
      COMMON/E/XX(150),YY(150),ALFA(150),IBLAD(150)/B/IHULLA(150,2)
      COMMON/C/IBANK(5),NUM(50),X(150),Y(150),DM(600)
      DIMENSION  W(30),Z(30)
      DATA (W(I),I=1,26)/1470.,1432.,1428.,1423.,1389.,1343.,1336.,1295.
     *,1280.,1280.,1265.,1252.,1244.,1232.,1217.,1204.,1192.,1177.,1159.
     *,1080.,936.,904.,900.,915.,950.,954./
      DATA (Z(I),I=1,26)/538.,570.,620.,664.,696.,758.,792.,900.,956.,
     *1020.,1050.,1112.,1126.,1175.,1235.,1288.,1313.,1400.,1465.,1537.,
     *1826.,1930.,1998.,2078.,2188.,2218./
      KK=KX
      IF (KX .LT. 0) KK=0
      CALL NAMPLT
      CALL YLIMIT(35.0)
      CALL XLIMIT(30.0)
      W(27)=Z(27)=10.0
      W(28)=Z(28)=SCAL
      CALL LINE(W,Z,26,1,0,3)
      DO 2 I=1,NBANK
      II=IBANK(I)
      X(I)=XX(II)
      Y(I)=YY(II)
    2 CONTINUE
      X(NBANK+1)=Y(NBANK+1)=10.0
      X(NBANK+2)=Y(NBANK+2)=SCAL
      CALL LINE(X,Y,NBANK,1,-1,3)
      ISTAR=1
      IEND=0
      DO 5 I=1,NTRUCK
      IEND=IEND+NUM(I)
      DO 4 II=ISTAR,IEND
      J=IHULLA(II,KK+1)
      JJ=II+1-ISTAR
      X(JJ)=XX(J)
      Y(JJ)=YY(J)
    4 CONTINUE
      N1=NUM(I)+1
      X(N1)=X(1)
      Y(N1)=Y(1)
      X(N1+1)=Y(N1+1)=10.0
      X(N1+2)=Y(N1+2)=SCAL
      CALL LINE(X,Y,N1,1,1,26)
      ISTAR=ISTAR+NUM(I)
    5 CONTINUE
      CALL SETPEN(2)
      YAX=(1313-10)/SCAL
      XEND=(1192-10)/SCAL
      CALL PLOT(0.0,YAX,3)
      CALL PLOT(XEND,YAX,2)
      CALL SYMBOL( 0.2,YAX,0.10,10H7200 NORTH,0.2,10)
      YAX=(1126-10)/SCAL
      XEND=(1244-10)/SCAL
      CALL PLOT(0.0,YAX,3)
      CALL PLOT(XEND,YAX,2)
      CALL SYMBOL( 0.2,YAX,0.10,10H3200 NORTH,0.2,10)
```

```
      YAX=( 956-10)/SCAL
      XEND=(1280-10)/SCAL
      CALL PLOT(0.0,YAX,3)
      CALL PLOT(XEND,YAX,2)
      CALL SYMBOL( 0.2,YAX,0.10,10H 800 SOUTH,0.2,10)
      YAX=( 761-10)/SCAL
      XEND=(1343-10)/SCAL
      CALL PLOT(0.0,YAX,3)
      CALL PLOT(XEND,YAX,2)
      CALL SYMBOL( 0.2,YAX,0.10,10H 500 SOUTH,0.2,10)
      YAX=( 538-10)/SCAL
      XEND=(1410-10)/SCAL
      CALL PLOT(0.0,YAX,3)
      CALL PLOT(XEND,YAX,2)
      CALL SYMBOL( 0.2,YAX,0.10,11H11000 SOUTH,0.2,11)
      XAX=(1265-10)/SCAL
      YEND=(1050-10)/SCAL
      CALL PLOT(XAX,0.0,3)
      CALL PLOT(XAX,YEND,2)
      CALL SYMBOL(XAX,0.0,0.10,6H0 WEST,90.0,6)
      XAX=(1080-10)/SCAL
      YEND=(1537-10)/SCAL
      CALL PLOT(XAX,0.0,3)
      CALL PLOT(XAX,YEND,2)
      CALL SYMBOL(XAX,0.0,0.10,9H4000 WEST,90.0,9)
      XAX=( 938-10)/SCAL
      YEND=(1826-10)/SCAL
      CALL PLOT(XAX,0.0,3)
      CALL PLOT(XAX,YEND,2)
      CALL SYMBOL(XAX,0.0,0.10,9H7500 WEST,90.0,9)
      CALL SYMBOL(0.2,1.0,0.1,20HMETROPOLITAN CHICAGO,0.2,20)
      CALL SYMBOL(0.2,0.5,0.1,20HINTER-HOSPITAL BLOOD,0.2,20)
      CALL SYMBOL(0.2,0.2,0.1,22HTRANSPORTATION NETWORK,0.2,22)
      YSTAR=2265/SCAL
      IF (KX .EQ. 0) CALL SYMBOL(0.2,YSTAR,0.1,40HALLOCATION BASED ON EM
     1ERGENCY COSTS ONLY,0.0,40)
      IF (KX .EQ. 1) CALL SYMBOL(0.2,YSTAR,0.1,38HALLOCATION BASED ON RO
     1UTING COSTS ONLY,0.0,38)
      IF (KX .EQ.-1) CALL SYMBOL(0.2,YSTAR,0.1,50HALLOCATION BASED ON EM
     1ERGENCY,ROUTING,SYSTEM COSTS,0.0,50)
      CALL ENDPLT
      RETURN
      END
```

```
      INTEGER FUNCTION DATX(I,J)
      DATX=(J-1)*150+I-1
      RETURN
      END
```