The Travelling Salesman Problem: Solution by a method of ranking assignments

Katta G. Murty *

Indian Statistical Institute, Calcutta, India

Caroline Karel

Case Institute of Technology, Cleveland, Ohio

John D. C. Little

Massachusetts Institute of Technology, Cambridge, Mass.

September 1962.

---

* On a study assignment at Case Institute of Technology, Cleveland when this work was done.

Abstract

Any tour of a travelling salesman problem is an assignment (in the sense of the assignment problem) but an assignment is not necessarily a tour. If the assignments of a travelling salesman problem are ranked in increasing order of the objective function, starting with the optimal assignment, then the first assignment which is a tour will be the optimal tour.

A procedure is given for finding the sequence of ranked assignments. Then a modified procedure is developed which finds subsequence which also leads to the optimal tour but with much greater computational efficiency. This is applied to an asymmetric 10 city problem having randomly chosen cost elements. After adding a modification to take advantage of symmetry, Croes' 20 city problem is solved.

Definitions and Notation

Consider any point $X = (x_{ij})$ satisfying the relations

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n$$

$$x_{ij}^2 = x_{ij} \qquad i, j = 1, \ldots, n$$

Thus X may be thought of as an n X n matrix with elements either 0 or 1. The word, assignment will be used to mean the set of x_{ij}'s which equal unity at the point X. Hence the statement "a particular assignment contains x_{25}" means that $x_{25} = 1$ at the point represented by the assignment.

The objective function is the linear form

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

where $C = (c_{ij})$ is the cost matrix.

Sometimes (i,j) will be used to denote x_{ij} and c (i,j) to denote c_{ij}.

Y denotes the set of all possible assignments for the problem concerned. y denotes any element of Y. z[y] is the value of the objective function at y

Y (r), where r is any index, is used to denote subsets of Y. Y(r,s), where s is another index, denotes a subset of Y(r), etc.

y(r), y(r,s) etc. denote the minimal assignments, i.e. the one which minimises the objective function, in the sets Y(r), Y(r,s) etc. respectively.

Assignment problem means the problem of finding the minimal (optimal) assignment. This may be done, for example, by using Flood's technique (the Hungarian method).

Flood's technique involves the modification of the cost matrix by adding constants to its rows and columns, until a matrix with all nonnegative elements and containing at least one zero in each row and column is obtained. This modified matrix at the end of the working by Flood's algorithm will be called the reduced cost matrix. The optimal assignment contains cells, one in each row and column, all of which correspond to zero cost values in the reduced cost matrix.

In reference to any assignment problem and an optimal assignment for it obtained by Flood's technique, $B_{i.}$ and $B_{.j}$ denote the minimal elements in the i th row and j th column respectively of the reduced cost matrix after excluding the cells contained in the optimal assignment. Define

$$\theta(ij) = B_{i.} + B_{.j}$$

Let $y = [(i_1,j_1),\ldots,(i_n,j_n)]$ be any assignment for a problem of size n. Then $[(j_1,i_1),\ldots,(j_n,i_n)]$, which is also an

assignment, will be called the transpose of y and denoted by R(y). When the cost matrix is symmetric both the assignment and its transpose yield the same value of z. Let Y(s) be any set of assignments. $R\{Y(s)\}$ will denote the set of all assignments which are transposes of those in Y(s).

A(0), A(1), ... denote the finite sequence of all assignments in Y arranged in increasing order of the objective function. While arranging the elements of Y in the form of this sequence, if at any stage several y have the same value of z. They should all be numbered serially in some order before counting the assignment with the next higher value of z.

Thus the equation

$$A(s) = \text{minimal assignment amongst } \{Y - \{A(0), \ldots, A(s-1)\}\} - (1)$$

is true for s = 1, 2, ...

Let α stand for any arbitrarily chosen very large positive number.

A Procedure for ranking assignments in increasing order of the objective function

The general idea is as follows:

Consider stage s. A(0), ..., A(s-1) are known and A(s) is to be found. At this stage Y is expressed as a disjoint union of

$$\{A(0)\}, \ldots, \{A(s-1)\}, Y(1), \ldots, Y(N)$$

where for r = 1, ..., N each of Y(r) has the property that y(r) can be found by the solving of a single assignment problem.

Then

$$A(s) = \text{minimal assignment among } \{Y-\{A(0), \ldots, A(s-1)\}\}$$
$$= \text{minimal assignment among } \{y(1), \ldots, y(N)\}$$

Now suppose this is done and A(s) = y(t). Then Y(t) is partitioned into disjoint subsets, say,

$$\{y(t)\} = \{A(s)\}, Y(t,1), \ldots, Y(t,M)$$

such that for r = 1, ..., M each of y(t,r) can be found by solving a single assignment problem. Then Y is the disjoint union of

$$\{A(0)\}, \ldots, \{A(s)\}, Y(1), \ldots, Y(t-1), Y(t+1), \ldots Y(N),$$
$$Y(t,1), \ldots, Y(t,M).$$

This disjoint decomposition of Y is of the same form as that at the beginning of stage s, except that it is advanced to s + 1.

To start the procedure A(0), the minimal assignment in Y, is found.

The details of the disjoint partitioning are now presented. Let the minimal assignment, (or any one of the minimal assignments if there is a tie), be

$$A(0) = [(i_1, j_1), \ldots, (i_n, j_n)]$$

let

$$Y(0) = \{A(0)\}$$

$$Y(1) = \{\text{all } y \text{ such that } (i_1, j_1) \notin y\}$$

$$Y(r) = \{\text{all } y \text{ such that } (i_1, j_1), \ldots, (i_{r-1}, j_{r-1}) \in y\}$$
$$\text{and } (i_r, j_r) \notin y$$
$$r = 2, \ldots, n-1.$$

It is seen that the Y(r) are mutually disjoint.

In an n X n problem no two assignments can contain more than n-2 common elements. Hence the only assignment containing

$$(i_1, j_1), \ldots, (i_{n-1}, j_{n-1}) \text{ is } A(0).$$

$$Y = \bigcup_{r=0}^{n-1} Y(r)$$

Now

$$A(1) = \text{minimal assignment among } \{Y - \{A(0)\}\}$$
$$= \text{minimal assignment among } \{y(1), \ldots, y(n-1)\} \quad (2)$$

y(1) is the solution of an assignment problem with a cost matrix (K_{ij}) in which $K(i_1, j_1) = \infty$ and all other $K_{ij} = C_{ij}$, since it is the minimal assignment not containing (i_1, j_1).

y(2) contains (i_1, j_1). The remaining elements of y(2) are got by solving the assignment problem of order n-1 with a cost matrix which is obtained by changing $C(i_2, j_2)$ to ∞ and then striking off the i_1th row and j_1th column from C.

Continuing, for r = 3, ..., n-1 y(r) is obtained similarly as the

solution of a single assignment problem of order n - r + 1. Hence A(1) can be obtained from (2). The computations required are the solving of n-1 different assignment problems in all, of which there is one of order r for r = 2, ..., n.

Note that if Y contains other minimal assignments besides A(0), A(1) will be one of them because A(1) is obtained from equation (2). In general, for s = 1, 2, ... since the procedure obtains A(s) from equation (1) it will enumerate all the assignments having a given value of the objective function before going on to the next higher value.

To find A(2), suppose A(1) = y(t) = $[(i_1, j_1), \ldots, (i_{t-1}, j_{t-1}); (m_t, p_t)\ldots(m_n, p_n)]$ where $(i_1, j_1), \ldots, (i_{t-1}, j_{t-1})$ are contained in all $y \in Y(t)$, and $(m_s, p_s) \neq (i_t, j_t)$ for s = t, ..., n by the definition of Y(t).

Now Y(t) will be expressed as a disjoint union of $\{y(t)\}$ and several other subsets, the minimal assignment in each of which is obtained as the solution of a single assignment problem.

Let

$$Y(t,0) = \{y(t)\}$$

$$Y(t,1) = \{\text{all } y \in Y(t) \text{ such that } (m_t, p_t) \notin y$$

$$Y(t,2) = \{\text{all } y \in Y(t) \text{ such that } (m_t, p_t) \in y,\ (m_{t+1}, p_{t+1}) \notin y$$

.....

$$Y(t, n-t) = \{\text{all } y \in Y(t) \text{ such that } (m_t, p_t), \ldots, (m_{n-2}, p_{n-2}) \in y \text{ and } (m_{n-1}, p_{n-1}) \notin y\}$$

Then

$$Y(t) = \bigcup_{s=0}^{n-t} Y(t,s)$$

$$\{Y(t) - \{y(t)\}\} = \bigcup_{s=1}^{n-t} Y(t,s)$$

and the required disjoint partition of Y(t) has been achieved.

$$\{Y - \{A(0), A(1)\}\} = \bigcup_{\substack{r=1 \\ \neq t}}^{n-1} Y(r) \bigcup_{s=1}^{n-t} Y(t,s).$$

$$\begin{aligned} A(2) &= \text{minimal assignment among } \{Y - \{A(0), A(1)\}\} \\ &= \text{minimal assignment among } \{y(1), \ldots, y(t-1), y(t+1), \ldots, \\ &\qquad y(n-1), \quad y(t,1), \ldots, y(t,n-t)\} \end{aligned} \tag{3}$$

As before, it is evident that each of y(t,s) for s = 1, ..., n-t is obtained as the solution of a single assignment problem. Hence A(2) is easily obtained from (3). The computations required are the solving of n-t different assignment problems, of which there is one each of orders r = 2, ..., n-t+1.

Repeated application of this procedure gives the sequence A(s) to any desired extent.

While computing for A(s) if several (say r+1) assignments show up having the same value of z, call them A(s), ..., A(s+r) and in order to continue the search for A(s+r+1), partition each of the r subsets from which these assignments came, in the manner described above.

At each stage the maximum amount of computations that may be required for getting the next A(s) are the solving of at most n-1 different assignment problems, one each of size 2, ..., n.

The Travelling Salesman Problem

This famous problem may be stated as follows.

A salesman wishes to visit n given cities in a single town. Starting from any one of them and visiting each of the others once and only once in some order, and returning to the starting point in the end. The costs of travelling from each city to each other are given. Obviously the total cost of the tour depends on the sequence in which the cities are visited. The problem is to develop an efficient algorithm for finding out the sequence associated with the minimal total tour cost.

Held and Karp [10] say "while no completely acceptable computational method exists for solving the travelling salesman problem, several procedures have been developed for obtaining optimum or near optimum solutions. These procedures, however, are usually somewhat tedious, intuitive, and difficult to program for a computer. A "state of the art" discussion of the travelling salesman problem may be found in [1]".

A dynamic programming formulation by Held and Karp yields a direct solution of problems of moderate size (13 or fewer cities) on an IBM 7090. Larger problems are treated by means of a "successive approximation technique" which may not always converge to the optimal tour.

Formulation of the Travelling Salesman Problem

Let C_{ij} be the cost of travelling from city i to city j. The fact that the salesman has to proceed to another city every time will be expressed by setting

$$C_{ij} = \infty \qquad i = 1, \ldots, n.$$

Let $(i,j) = 1$ if salesman proceeds from city i to city j

$= 0$ otherwise

Let D_n be the set of all possible permutations of the integers $1, \ldots, n$. Any element $\alpha \in D_n$ is an ordered vector $(t_1, t_2, \ldots, t_n)$ which is a permutation of the integers $1, \ldots, n$. To each element $\alpha = (t_1, \ldots, t_n) \in D_n$ there can be associated an assignment:

$$[(t_1, t_2), (t_2, t_3), \ldots., (t_{n-1}, t_n), (t_n, t_1)]$$

Such an assignment will be called a tour. Let T be the set of all hours. For any fixed n it is easily seen that $T \subset Y$.

Any assignment which is

(1) not a tour and

(2) does not include any of the elements of the principal diagonal, i.e., (i,i) $i = 1, \ldots, n$.

will be called a nontour.

Let U_n = totalnumber of tours and non tours for the n city problem. Since r out of the n cells along the principal diagonal can be selected in $\binom{n}{r}$ ways

$$\binom{n}{r} U_{n-r}$$

gives the number of assignments which contain exactly r cells among the principal diagonal.

Thus U_n is given by

$$\binom{n}{o} U_n + \ldots + \binom{n}{n-1} U_1 + \binom{n}{n} = n!$$

with the boundary conditions $U_1 = 0$, $U_2 = 1$.

Hence $\{U_n - (n-1)!\}$ gives the number of possible nontours in the n-city problem.

An assignment which can be arranged in the form

$$[(t_1, t_2), (t_2, t_3), \ldots (t_m, t_1), \ldots.]$$

with $m < n$ is said to contain a subtour of length m and must be a non tour.

Let T(0) be the optimal tour, i.e. the minimal assignment among T.

Solving the Travelling Salesman Problem

Since $T \subset Y$, the procedure of ranking assignments in increasing order of the objective function can be used to find the optimal tour. Using the cost matrix C develop the sequence A(s) until an assignment is found which is also a tour. This will be the optimal tour.

However, the computations required can be greatly reduced by omitting on the way several assignments which cannot possibly be tours. A sequence B(s) would thus be obtained, which is a subsequence of A(s), leading to the optimal tour.

The procedure is as follows. Consider stage s. At the end of the previous stage Y has been partitioned into disjoint subsets

$\{B(0)\}, \ldots, \{B(s-1)\}, Y(1), \ldots, Y(N)$ and some discarded subsets which have been ascertained to consist entirely of non-tours.

$y(\delta)$ has either been obtained in the previous stages of the procedure, or may be obtained as the solution of a single assignment problem, $\delta = 1,2,\ldots,N$.

Set

$$B(s) = \text{minimal assignment among } \{y(1), \ldots, y(N)\}$$
$$= y(r), \text{ say.}$$

In general Y(r) will be of the form

$$Y(r) = \{\text{all } y \in Y \text{ such that } (i_1,j_1)\ldots(i_e,j_e) \in y \text{ and } (i_{e+1},j_{e+1}), \ldots, (i_{e+k},j_{e+k}) \notin y\}$$

and y(r) will be of the form

$$y(r) = [(i_1,j_1)\ldots(i_e,j_e); (m_1,p_1)\ldots(m_w,p_w)\ldots(m_{n-e},p_{n-e})]$$

Further partitioning of Y at this stage, if required, is performed according to the following steps.

Step 1: If B(s) = y(r) is a tour, it is the optimal tour, and the procedure has been completed.

Otherwise find the subtour in B(s) = y(r) which contains the smallest number of elements of the type (m,p), i.e. elements not already forced into y(r) by the definition of Y(r). Let w be the number of (m,p) elements in such a subtour.

Step 2: If there is more than one subtour in B(s) containing w of (m,p) elements, choose that subtour among them which maximises

$$\text{minimum for } (m,p) \text{ in subtour } [\theta(m,p)]$$

where the θ values are calculated from the reduced cost matrix which gave y(r) = B(s).

Step 3: Having selected the subtour, index the (m,p) elements in it in order of decreasing θ so that (m_1,p_1) has the highest θ value and (m_w,p_w), the smallest.

Step 4: Partitioning Y(r) let

$$Y(r,1,0) = \{\text{all } y \ Y(r) \text{ such that } (m_1,p_1) \notin y\}$$

$$Y(r,2,0) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1) \in y,\ (m_2,p_2) \notin y \text{ and also } (p_1,m_1) \notin y\}$$

$$Y(r,3,0) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1)\ (m_2,p_2) \in y,\ (m_3,p_3) \notin y \text{ and also } (p_1,m_1)\ (p_2,m_2) \notin y\}$$

.
.
.

$$Y(r,w,0) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1)\ldots(m_{w-1},p_{w-1}) \in y,\ (m_w,p_w) \notin y \text{ and also } (p_1,m_1)\ldots(p_{w-1},m_{w-1}) \notin y\}$$

Step 5: the retained subsets in the partition of Y are the disjoint sets

$$\{B(0)\}\ldots\{B(s-1)\},\{B(s)\},\ Y(1)\ldots Y(r-1),\ Y(r+1)\ldots Y(N),\ Y(r,1,0)\ldots.Y(r,w,0).$$

and the procedure is now returned to the beginning to commence stage s + 1.

Proof that no tours are discarded from Y in step 4, 5:

Let

$$Y(r,1) * Y(r,1,0)$$

$$Y(r,2) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1) \in y,\ (m_2,p_2) \notin y\}$$

$$Y(r,3) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1),\ (m_2,p_2) \in y,\ (m_3,p_3) \notin y\}$$

.
.
.

$$Y(r,w) = \{\text{all } y \in Y(r) \text{ such that } (m_1,p_1)\ldots(m_{w-1},p_{w-1}) \in y,\ (m_w,p_w) \notin y\}$$

If

$$y \in \{Y(r) - \{y(r)\}\} \bigcup_{\delta=1}^{w} Y(r,\delta)\}$$

then $(m_1,p_1)....(m_w,p_w)$ are all $\in y$ and hence y contains a subtour and so is a nontour.

Further, for $\delta = 2,3, \ldots, w$, it is obvious from the definition of $Y(r,\delta,0)$ that

$$Y(r,\delta,0) \subset Y(r,\delta)$$

and that any assignment in $\{Y(r,\delta) - Y(r,\delta,0)\}$ contains within it, at least one subtour of the form (m_u,p_u) (p_u, m_u), and is therefore a nontour.

Thus $\{Y(r) - \{y(r)\} \bigcup_{\delta=1}^{w} Y(r,\delta,0)$

consists entirely of nontours and may therefore be completely discarded from further computations in searching for the optimal tour. This justifies step 5.

Motivation for steps 1, 2, 3:

By these steps the procedure attempts to make

$$z[B(s+1)] - z[B(s)]$$

as large as possible, while keeping the number of assignment problems to be solved at each stage as low as possible.

Suppose the optimal assignment with respect to any cost matrix $\llcorner$, contains the cell (i,j) and yields a value of z_o for the corresponding objective function. Then from the definition of $\theta(i,j)$ it is obvious that

$$z \begin{bmatrix}\text{minimal assignment not containing} \\ (i,j) \text{ for cost matrix } \llcorner\end{bmatrix} \geq z_o = \theta(i,j)$$

Due to this, step 2 assures that the sequence z[B(s)] would jump at least by the indicated max min [θ(m,p)], thus achieving the purpose of securing the largest possible jump.

The indexing in step 3 is an attempt to secure the highest value of z for the minimal assignments among the subsets containing the largest number of assignments. This has an effect of reducing the sizes of assignment problems to be solved in succeeding stages.

Above all, step 1, by choosing the subtour in B(s) containing the smallest number of elements of the type (m,p); keeps the number of assignment problems to be solved for obtaining B(s + 1) as low as possible.

Commencing the procedure, Stage 1:

The proceedure is started by finding the minimal assignment in Y, which is = A(0) = B(0).

A 10 City Asymmetric example

A cost matrix obtained by taking c(i,j) from a table of random numbers is:

$$\begin{bmatrix}
\alpha & 51 & 55 & 90 & 41 & 63 & 77 & 69 & 0 & 23 \\
50 & \alpha & 0 & 64 & 8 & 53 & 0 & 46 & 73 & 72 \\
30 & 77 & \alpha & 21 & 25 & 51 & 47 & 16 & 0 & 60 \\
65 & 0 & 6 & \alpha & 2 & 9 & 17 & 5 & 26 & 42 \\
0 & 94 & 0 & 5 & \alpha & 0 & 41 & 31 & 59 & 48 \\
79 & 65 & 0 & 0 & 15 & \alpha & 17 & 47 & 32 & 43 \\
76 & 96 & 48 & 27 & 34 & 0 & \alpha & 0 & 25 & 0 \\
0 & 17 & 9 & 27 & 46 & 15 & 84 & \alpha & 0 & 24 \\
56 & 7 & 45 & 39 & 0 & 93 & 67 & 79 & \alpha & 38 \\
30 & 0 & 42 & 56 & 49 & 77 & 72 & 49 & 23 & \alpha
\end{bmatrix}$$

Commencing stage 1, it was found that the optimal assignment is

$$B(0) = A(0) = [(10,2)\ (7,10)\ (2,7),\ (1,9)\ (9,5)\ (5,6)\ (6,4)\ (4,3)\ (3,8)\ (8,1)]$$

giving

$$z[B(0)] = 22$$

Commencing stage 2, it is found from B(0) that $w = 3$. Defining

$$Y(1) = \{\text{all } y \in Y \text{ such that } (10,2) \notin y\}$$

$$Y(2) = \{\text{all } y \in Y \text{ such that } (10,2) \in y,\ (7,10)\ (2,10) \notin y\}$$

$$Y(3) = \{\text{all } y \in Y \text{ such that } (10,2)\ (7.10) \in y,\ (2,7)\ (2,10)\ (10,7) \notin y\}$$

Then

$$B(1) = \text{minimal assignment among } \{y(1), y(2), y(3)\}$$

$$= y(2) = [(10,2);\ (9,5)\ (3,9)\ (5,3),\ (2,7)\ (7,6)\ (6,4)\ (4,8)\ (8,1)\ (1,10)]$$

with $z[B(1)] = 28$

Commencing stage 3, define

$$Y(4) = \{\text{all } y \in Y(2) \text{ such that } (9,5) \notin y\}$$

$$Y(5) = \{\text{all } y \in Y(2) \text{ such that } (9,5) \in y,\ (3,9)\ (5,9) \notin y\}$$

$$Y(6) = \{\text{all } y \in Y(2) \text{ such that } (9,5)\ (3,9) \in y,\ (5,3)\ (5,9)\ (9,3) \notin y\}$$

and $B(2) = \text{minimal assignment among } \{y(1), y(3), y(4), y(5), y(6)\} = y(6)$

$$= [(10,2)\ (9,5)\ (3,9);\ (6,4)\ (4,3)\ (5,6),\ (2,7)\ (7,8)\ (8.1)\ (1,10)].$$

with $z[B(2)] = 29.$

Commencing stage 4, let

$$Y(7) = \{\text{all } y \in Y(6) \text{ such that } (6,4) \notin y\}$$

$$Y(8) = \{\text{all } y \in Y(6) \text{ such that } (6,4) \in y,\ (4,3)\ (4,6) \notin y\}$$

$$Y(9) = \{\text{all } y \in Y(6) \text{ such that } (6,4)\ (4,3) \in y,\ (5,6)\ (4,6)\ (3,4) \notin y\}$$

and $B(3) = \text{minimal assignment among } \{y(1), y(3), y(4), y(5), y(7), y(8), y(9)\}$

$$= y(7)$$

$$= [(10,2)\ (9,5)\ (3,9);\ (5,4)\ (4,8)\ (8,1)\ (1,10)\ (2,7)\ (7,6)\ (6,3)]$$

which is a tour and hence the optimal tour with

$$z[B(3)] = 33.$$

Continuing the procedure in similar manner, a stage further, it was found that the optimal tour is unique.

Special case of a symmetric cost matrix:

When the cost matrix is symmetric any assignment and its transpose yield the same value for the objective function. Thus the transpose of the optimal tour would be another optimal tour. This fact, when used in the computational procedure for the travelling salesman problem, permits at any stage, the discarding of not only the partitions containing all nontours, but also those whose transposes form subsets of some of the other partitions which are retained for the next stage of the algorithm. This modification to the procedure can usually be expected to effect considerable computational savings.

In a symmetric problem the first few assignments in the sequence B(0), B(1), ... will usually contain several two element subtours. Suppose that

$$B(0) = A(0) = [(i,j)\ (j,i)\ .....]$$

and that the subtour (i,j) (j,i) is the one selected to be used for partitioning Y. Then the subsets to be retained for the next stage would normally be

$$Y(1) = \{\text{all } y \in Y \text{ such that } (i,j) \notin y\}$$

$$Y(0,2) = \{\text{all } y \in Y \text{ such that } (i,j) \in y,\ (j,i) \notin y\}$$

But

$$R\{Y(0,2)\} = \{\text{all } y \in Y \text{ such that } (j,i) \in y,\ (i,j) \notin y\}$$

and therefore

$$R\{Y(0,2)\} \subset Y(1).$$

Thus it suffices to consider only Y(1) for the succeding stages, since even if Y(0,2) were to contain the optimal tour, its transpose, which is also optimal, would be contained in Y(1). Thus

$$B(1) = y(1).$$

Some additional gain can be made at the second stage if a subtour of two elements appears in B(1) also. Suppose

$$B(1) = y(1) = [(m,p)\ (p,m),\ \ldots.]$$

The normal partition of Y(1) would be

$$Y(1,1) = \{\text{all } y \in Y \text{ such that } (i,j)\ (m,p) \notin y\}$$

$$Y(1,2) = \{\text{all } y \in Y \text{ such that } (m,p) \in y;\ (i,j)\ (p,m) \notin y\}$$

and one would usually retain only these two subsets for the third stage. However let

$$Y(1,2,0) = \{\text{all } y \in Y \text{ such that } (j,i)\ (m,p) \in y \quad (i,j)\ (p,m) \notin y\}$$

$$Y(1,2,1) = \{\text{all } y \in Y \text{ such that } (m,p) \in y\ (i,j)\ (p,m)\ (j,i) \notin y\}$$

Then $Y(1,2) = Y(1,2,0) \cup Y(1,2,1)$

and $R\ Y(1,2,1) \subset Y(1,1)$.

Thus it suffices to consider only the subsets Y(1,1) and Y(1,2,0) for the third stage.

Application of this principle beyond the second stage, even if two element subtours persist among the B(s) in the sequence, does not appear to be very fruitful.

A 20 City Symmetric Example:

The algorithm together with the simplyications applicable under the special case of symmetric cost matric, has been used to solve the 20 city problem due to Crola [7].

The optimal assignment for the problem is

$$B(0) = A(0) = [(16,8)\ (8,16),\ (1,12)\ (12,1)\ (2,14)\ (14,2)\ (3,20)\ (20,3)\ (4,13)\ (13,4)\ (5,9)\ (9,5)\ (6,19)\ (19,6)\ (7,15)\ (15,7)\ (10,18)\ (18,10)\ (11,17)\ (17,11)]$$

with $z[B(0)] = 218$,

where (16,8) (8.16) form a subtour of two elements within B(0), both corresponding to the maximum value of $\theta = 17$.

Commencing stage 2, let

$$Y(1) = \{\text{all } y \in Y \text{ such that } (16,8) \notin y\}$$

and according to the algorithm

$$B(1) = y(1) = [(9,5)\ (5,9),\ (1,12)\ (12,1)\ (2,14)\ (14,2)\ (3,20)\ (20,3)\ (4,13)\ (13,4)\ (10,18)\ (18,10)\ (11,17)\ (17,11)\ (6,19)\ (19,7)\ (7,15)\ (15,8)\ (8,16)\ (16,6)]$$

with $z[B(1)] = 235$.

Commencing stage 3, let

$$Y(2) = \{\text{all } y \in Y(1) \text{ such that } (9,5) \notin y\}$$

$$Y(3) = \{\text{all } y \in Y(1) \text{ such that } (8,16)\ (9,5) \in y,\ (5,9) \notin y$$

and define

$$B(2) = \text{minimal assignment among } \{y(2),\ y(3)\}$$

It was found that Y(2) has two optimal assignments and that both these and y(3) yield the same value of z. However all these are nontours and it is necessary to continue the algorithm further.

As has been noted under the procedure for developing the sequence of ranked assignments, any one of the optimal assignments may be considered first, and the other one would automatically turn up in the succeeding stage of computation. Thus let

$$B(2) = \text{one of the optimal assignments in } Y(2)$$

$$= [(1, 12)\ (12,1),\ (11,17)\ (17,11)\ (13,4)\ (4,13)$$
$$(2,8)\ (8,16)\ (16,15)\ (15,7)\ (7,19)\ (19,6)\ (6,5)$$
$$(5,9)\ (9,3)\ (3,20)\ (20,18)\ (18,10)\ (10,14)\ (14,2)]$$

$$B(3) = y(3) = [(1,16)\ (9,5);\ (19,7)\ (7,19),\ (16,15)$$
$$(15,8)\ (1,4)\ (4,13)\ (13,2)\ (2,14)\ (14,10)$$
$$(10,18)\ (18,20)\ (20,3)\ (3,9)\ (5,6)$$
$$(6,17)\ (17,11)\ (11,12)\ (12,1)].$$

with $z[B(2)] = z[B(3)] = 245.$

In stage 4 it is necessary to partition both Y(2) and Y(3) in order to get the next element in the sequence B(s). Let

$$Y(4) = \{\text{all } y \in Y(2) \text{ such that } (1,12) \notin y\}$$
$$Y(5) = \{\text{all } y \in Y(2) \text{ such that } (1,12) \in y,\ (12, 1) \notin y\}$$
$$Y(6) = \{\text{all } y \in Y(3) \text{ such that } (19,7) \notin y\}$$
$$Y(7) = \{\text{all } y \in Y(3) \text{ such that } (19,7) \in y,\ (7,19) \notin y\}$$

It was found that Y(2) has two optimal assignments and that both these and y(3) yield the same value of z. However all these are nontours and it is necessary to continue the algorithm further.

As has been noted under the procedure for developing the sequence of ranked assignments, any one of the optimal assignments may be considered first, and the other one would automatically turn up in the succeeding stage of computation. Thus let

$$B(2) = \text{one of the optimal assignments in } Y(2)$$
$$= [(1, 12)\ (12,1),\ (11,17)\ (17,11)\ (13,4)\ (4,13)$$
$$(2,8)\ (8,16)\ (16,15)\ (15,7)\ (7,19)\ (19,6)\ (6,5)$$
$$(5,9)\ (9,3)\ (3,20)\ (20,18)\ (18,10)\ (10,14)\ (14,2)]$$

$$B(3) = y(3) = [(1,16)\ (9,5);\ (19,7)\ (7,19),\ (16,15)$$
$$(15,8)\ (1,4)\ (4,13)\ (13,2)\ (2,14)\ (14,10)$$
$$(10,18)\ (18,20)\ (20,3)\ (3,9)\ (5,6)$$
$$(6,17)\ (17,11)\ (11,12)\ (12,1)].$$

with $z[B(2)] = z[B(3)] = 245.$

In stage 4 it is necessary to partition both Y(2) and Y(3) in order to get the next element in the sequence B(s). Let

$$Y(4) = \{\text{all } y \in Y(2) \text{ such that } (1,12) \notin y\}$$
$$Y(5) = \{\text{all } y \in Y(2) \text{ such that } (1,12) \in y,\ (12,1) \notin y\}$$
$$Y(6) = \{\text{all } y \in Y(3) \text{ such that } (19,7) \notin y\}$$
$$Y(7) = \{\text{all } y \in Y(3) \text{ such that } (19,7) \in y,\ (7,19) \notin y\}$$

And then

$$B(4) = \text{minimal assignment among } \{y(4), y(5), y(6), y(7)\}$$

$$= y(5) = [(1,12); (19,7)\ (7,19),\ (8,16)\ (16,15)\ (15,8)$$
$$(12,11)\ (11,17)\ (17,6)\ (6,5)\ (5,9)\ (9,3)$$
$$(3,20)\ (20,18)\ (18,10)\ (10,14)\ (14,2)\ (2,13)$$
$$(13,4)\ (4,1)]$$

with $z[B(4)] = 245$.

It may be noted that $B(4) = y(5)$ is the other optimal assignment in $Y(2)$.

Going over to stage 5, let

$$Y(8) = \{\text{all } y \in Y(5) \text{ such that } (19,7) \notin y\}$$

$$Y(9) = \{\text{all } y \in Y(5) \text{ such that } (19,7 \in y,\ (7,19) \notin y\}$$

and then

$$B(5) = \text{minimal assignment among } \{y(4), y(6), y(7), y(8), y(9)\}.$$

It was found that $Y(7)$ has two optimal assignments and that both these and $y(4)$ yield the same value for z. However only one of the optimal assignments of $Y(7)$ is of interest to the problem since it happens to be a tour. Hence that is the optimal tour.

$$B(5) = \text{the tour which is one of the minimal assignments in } Y(7)$$

$$= \text{the optimal tour}$$

$$= [(8,16)\ (9,5)\ (19,7);\ (1,4)\ (4,13)$$
$$(13,2)\ (2,14)\ (14,10)\ (10,18)\ (18,20)$$
$$(20,3)\ (3,9)\ (5,19)\ (7,15)\ (15,8)$$
$$(16,1)\ (6,17)\ (17,11)\ (11,12)\ (12,1)].$$

with $z[B(5)] = 246$.

The transpose of $B(5)$, which is another optimal tour, is the solution obtained by Croes.

Computational Time:

The 10-city problem has been solved by hand, and the time taken was about half an hour.

The 20-city problem involved the solving of 10 different assignment problems of sizes ranging from 16 to 20. On the Burroughs 220 computer, at the Computing Centre, Case Institute of Technology, this took about 10 minutes in all.

Acknowledgement

The authors thank the Case Computing Centre for generously supplying computer time for this work.

References

(1). R. L. Ackoff (ed.), Progress in Operations Research, Vol 1, Wiley, New York, 1961.

(2). C. W. Churchman, R. L. Ackoff, E. L. Arnoff (ed.), Introduction to Operations Research, Wiley, New York, 1960.

(3). M. Sasieni, A. Yaspan, L. Friedman, Operations Research Methods and Problems, Wiley, New York, 1959.

(4). S. Vajda, Mathematical Programming, Addison - Wesley, Reading, Mass., 1961.

(5). W. W. Garvin, Introduction to Linear Programming, Mcgraw - Hill, New York, 1960.

(6). R. Bellman, Dynamic Programming treatment of the travelling salesman problem, J. Assoc. Comput. Mach., 9, (1962), pp 61-63.

(7). G. A. Croes, A Method for solving travelling salesman problems, Operations Research, 6, (1958) pp 791 - 812.

(8). G. B. Dantzig, D. R. Fulkerson, S. M. Johnson, Solution of a large scale travelling salesman problem, Operations Research, 2, (1954), pp 393 - 410, and On a linear programming combinatorial approach to the travelling salesman problem, Operations Research, 7, (1959), pp 58 - 66.

(9). M. M. Flood, The travelling salesman problem, Operations Research, 4, (1956), pp 61 - 75.

(10). M. Held, R. M. Karp, A Dynamic Programming approach to sequencing problems, J. Soc. Indust. Appl. Math., 1, (1962), pp 196 - 210.