Optimization Algorithm Applied to the Problem of Routes for Symmetric Matrix And Non Symmetric Matrix

1Camargo, M.E., 2Sortica, G., 3Russo, S.L., 4Fabris, J.P., 5Dullis, A. I. S., 6Gassen, I.M., 7Oliveira, M.A.

1University of Santa Cruz do Sul, Department of Mathematics, Santa Cruz do Sul, RS, Brazil.
2University of Caxias do Sul, Caxias do Sul, RS, Brazil.
3Federal University of Santa Maria, Santa Maria, RS, Brazil.
4Federal University of Sergipe, Post-Graduate Program in Intellectual Property Science, São Cristóvão, Aracaju, SE, Brazil
5Federal University of Santa Maria, Santa Maria, RS, Brazil.
6University of Santa Cruz do Sul, Department of Mathematics, Santa Cruz do Sul, RS, Brazil.
7University of Santa Cruz do Sul, Department of Mathematics, Santa Cruz Sul, RS, Brazil.

ABSTRACT

The route of a vehicle fleet of known capabilities, which must go to a central depot for a given number of delivery, may require the selection of a large number of routes, if the number of deliveries is great also, in this context it is necessary an algorithm that determines the best route. This work had as objective in addition to considering some theoretical aspects of the problem of routes for symmetric matrix and not symmetrical, develop an iterative procedure of branches and limits, based on the traveling salesman problem and adapted to the problem of routes for symmetric and symmetric matrix, which offers a great route. It was observed that the difficulty of obtaining the final solution grows with the number of vehicles used and the number of iterations depends on the dimension of the array, the homogeneity of its elements and the relationship between the sum of the demands and the sum of the capacities of vehicles used. The problems that use larger number of vehicles present greater difficulty to obtain the final solution, and therefore requires more computation time.

INTRODUCTION

An algorithm of branches and limits with use of penalties is introduced to solve traveling salesman problems and problems of transport routes, in which there are goods to be distributed in known quantities, to destinations also known.

These goods are transported by vehicle fleets of known capabilities, and must designate each vehicle a source and one or more targets such that each vehicle begins its journey at the source, delivers the goods in each designated point and returns to the origin the aim is to minimize the total cost of delivery. If the vehicles used for delivery of the goods are of different capacities, the wonderful combination must be found such that the total distance travelled (cost or time) is minimized.

When a seller initiates a journey of one of the "n" cities and visit each of the "n-1" remaining cities, one and only once and returns to home town, have multiple traveling salesman problem, which is an extension of the simple traveling salesman. In this case, are M and n cities, vendors and found M routes, such that, each city is visited by only once, by a single seller, such that the total distance travelled for each salesperson to be minimized (Bellmore; Nemharser, 1968).

This work is developed an algorithm that Eilon algorithm extension and others (Eilon et al., 1971), for solving symmetric and not symmetrical to the traveling salesman problem simple and multiple, as well as the problem of routes with vehicles of same capacity or different capabilities (Lawler, Wood, 1966; Moore, 1991; Sortica, 1982, Tsai et al, 2004).

MATERIAL AND METHODS

Case Study:

Was solved a problem with symmetric random numbers and of dimension 6 x 6, using 2 vehicles, same capacity and equal to 50 units for delivery. The demands of the 4 points considered, are random numbers like 12, 23, 25 and 10 respectively. Depending on model, the actual deposit is replaced by 2 artificial deposits represented by delivery
vehicles, and occupy the last two rows and columns of the array away.

The behavior of the algorithm was analyzed, using a sample of 10 vehicles with symmetric matrices whose elements were generated randomly through a subroutine RANDU, called by the main program. The generated numbers range from 1 to 99. Demands are random numbers and matrices of dimensions ranging from 7×7 to 15×15, with 5 of the problems using 2 vehicles and 5 vehicles using 3 for delivery, all of same capacity equal to 60 units. The number of iterations of each problem, to reach the final solution range from 17 to 1161.

A model application with actual data in a gas distribution company. The cities that must be carried out deliveries, were grouped into two regions, determining two symmetrical problems of dimensions 14 x 14 and 16 x 16. In problem 1, 16 x 16 matrix, 2 vehicles were used, due to the sum of the demands and the capabilities of vehicles. The final solution was obtained with 126 iterations. The problem 2, who used 3 vehicles, obtained the final solution with 3410 iterations.

It was observed that the difficulty of obtaining the final solution grows with the number of vehicles used and the number of iterations depends on the dimension of the array, the homogeneity of its elements and the relationship between the sum of the demands and the sum of the capacities of vehicles used. The problems that use larger number of vehicles present greater difficulty to obtain the final solution and therefore require greater computation time.

Development of Algorithm:
A - Theorems used:
1) if the constant fj is subtracted from all the integers of column j of the array D distance (or time) and the constant ei is subtracted from the ith row of the resulting matrix, then the great trip in final D matrix is a great trip in original matrix. Also, if Z(T) is the distance from a trip in the matrix end, then:

\[ Z'(t) = Z(t) - \sum f_j - \sum e_i \]

the numbers ei and fj are called reduced constants.

2) If a certain trip t does not contain a pair (h, k), Z(t) ≥ min_{j:h} d_{h,j} + min_{i:k} d_{i,k} = P_{hk} where P_{hk} is called the penalty for not include the pair (h, k).

This theorem is a consequence of that, if the trip does not contain (h, k), so it should contain the pairs (h, x), (y, k) where x is the smallest possible value for these pairs are:

\[ \min_{j:h} d_{h,j} \quad \text{and} \quad \min_{i:k} d_{i,k} \]

B - Algorithm:
The functioning of the algorithm will be explained through the flowchart steps of Figure 5.

1- Read the distance matrix D (cost or time), mark Z_0 (distance from the best trip yet) to infinity and enumerate the node X, which will continue to branch.

2- Reduce the distance matrix D and do q_0 sum of constants reduced, and V(X) (lower limit of node X) q_0.

3- Choose the node (h, k) named node Y, to branch, so that the greater penalty P_{hk}.

4- Branch to the node Y, and make your limit be \( V(Y) = V(X) + P_{hk} \).

5 - Branch to the node Y. Eliminate the row i and column k. Put in endless D to prevent subtr. Reduce D and find q (sum of the constants reduced) and make your limit be \( V(Y) = V(X) + q \).

6-7 - To route problems with capacity constraints of the vehicle, it is necessary to check the sum of the demands of pairs of cities selected so far to continue. Check if the sum is less than or equal to the capacity of the vehicle chosen. If it is, continue, otherwise do \( V(Y) = \infty \) in the last node branched and continue.

8-15-16 - If D is a 2 x 2 matrix there are only two possible pairs on the left, (i, j) and he completes the journey. If the distance from this trip \( V(Y) < Z_0 \), register the trip, adapt \( Z_0 \) and continue; otherwise, only continue.

9 - Select the next node X to branch, which has the lowest limit V(X).

10 - If every boundaries are smaller than \( Z_0 \), the branch is completed and the journey, at this point, is great and its distance is \( Z_0 \).

11- If the branch is to be continued from node Y, which was branched (i.e., if X=Y as in step 3) go to step 3 without building the matrix D. Otherwise, continue.

12-13- If the matrix is symmetric do \( d_{ij} = \infty \) for the corresponding node X in the matrix D, in order to avoid the formation of subtour; otherwise, only continue.

14- The matrix D should be reconstituted and established the corresponding to the node X, as follows: read the distance matrix D. Find R = \( \sum d_{ij} \) for all elements d_{ij} that were included in the said series, going from the top of the tree at node X. To each of these dij, delete the row i and column j of the matrix, inserting infinite the appropriate cells of the matrix D, in order to prevent subtr. For each node in the series, the top of the tree to X, put the corresponding element of the array D to infinity as well as make \( d_{ij} = \infty \) if the matrix is symmetric, in order to prevent subtr. Reduce the array D and form the new limit in X, such that \( V(X) = R \) sum of constants reduced.

Illustrative Example:
It was developed a symmetrical problem with random numbers and of dimension 6 x 6, using 2
vehicles, same capacity and equal to 50 units for delivery.

The demands of the 4 points considered, are random numbers like 12, 23, 10 and 25 respectively. Depending on model, the actual deposit is replaced by 2 artificial deposits represented by delivery vehicles, and occupy the last two rows and columns of the matrix.

The matrix is symmetric, then \( d_{ij} = d_{ji} \) and to avoid subtur, is \( d_{ii} = \infty \) to \( i=j \) and the designation between vehicles is avoided by making \( d_{N-1, N} = d_{N, N-1} = \infty \), as seen in Figure 1.

\[
\text{Fig. 1: Input Matrix of illustrative example}
\]

Thus, the objective function of the problem is:

\[
\text{Min } Z = \sum_{i=1}^{6} \sum_{j=1}^{6} d_{ij} \ X_{ij} = \infty \ X_{11} + 15X_{12} + 26X_{13} + 14X_{14} + 8X_{15} + 8X_{16} + \cdots + 8X_{61} + 11X_{62} + 15X_{63} + 9X_{64} + \infty X_{65} + \infty X_{66}.
\]

where \( d_{ij} = \infty \) for \( i=j \) and \( X_{ij} = 1 \) if the distributor will directly from \( i \) to \( j \); \( X_{ij} = 0 \) otherwise.

For clarity of the issue, are presented a few steps of the algorithm development on the same solution,

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & \text{Minor element } e_i \\
1 & \infty & 4 & 16 & 5 & 0 & 0 \\
2 & 7 & \infty & 2 & 10 & 3 & 3 \\
3 & 18 & 1 & \infty & 1 & 7 & 7 \\
4 & 6 & 8 & 0 & \infty & 1 & 1 \\
5 & 0 & 0 & 5 & 0 & \infty & 0 \\
6 & 0 & 0 & 5 & 0 & \infty & 0 \\
\end{array}
\]

\text{Fig. 2: Reduced matrix for column with constants of reduction}

where \( V(X) = q_0 = \sum f_j + \sum e_i = 57 \) for \( X = 1 \).

Figure 3 shows the first array with reduced penalties for each null element. The biggest penalty is \( P_{4,3} = 1 \) and branch is chosen for the pair (4,3), whose limit is given by \( V(\bar{Y}) = q_0 + q_1 = 57 + 1 = 58 \), where \( q_1 \) is the sum of the last small constants.

The sum of the demands of selected points is 35 < 50 and the branch must continue for \( Y \) being \( V(X) = V(2) = 58 \).

\[
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & \text{2nd Minor} \\
1 & \infty & 4 & 16 & 5 & 0 & 0 \\
2 & 5 & \infty & 0 & 8 & 1 & 1 \\
3 & 17 & 0 & \infty & 0 & 6 & 0 \\
4 & 6 & 8 & 0 & \infty & 1 & 1 \\
5 & 0 & 0 & 5 & 0 & \infty & 0 \\
6 & 0 & 0 & 5 & 0 & \infty & 0 \\
\end{array}
\]

\text{Fig. 3: First matrix with reduced penalties}
After the reductions made in Figure 3, the Figure 4 is obtained, whose biggest penalty is $P_{3,2} = 6$, the branch is made in (3.2) and the limit by not including this pair on the trip is given by $V(Y) = 58 + 6 = 64$, and deterrent in (2.4) to avoid a sub-trip with selected pairs: (4.3), (3.2), then (2, 4) and reduces the matrix again. The sum of the demands for these points is $58 > 50$ and branch gets interrupted $V(3) = \infty$, as shown in Figure 2.

The lower limit is $V(\overline{Y}) = 58$ of step $X=2$ and being symmetric matrix with branching of $\overline{Y}$, it is $(4.3) = \infty$ and $(3, 4) (3, 4) = \infty$ so that they are not selected. Reduces again the matrix and gets the new $V(X) = 57 + 1 = 58$ to $X=2$. The biggest penalty is $P_{3,2} = 6$ and the limit $V(\overline{Y}) = 58 + 6 = 64$ as can be seen in Figure 2.

Infinite arises in the cell (2, 3) as well as other necessary deterrents, as has already been described and continue until you find a $2 \times 2$ matrix getting the best determined sequence of the trip.

### Table 1: Results of the problem

<table>
<thead>
<tr>
<th>Route</th>
<th>Route 1</th>
<th>Route 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of vehicles</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Distance travelled</td>
<td>50 units</td>
<td>18 units</td>
</tr>
<tr>
<td>Load</td>
<td>45 units</td>
<td>25 units</td>
</tr>
<tr>
<td>Sequence</td>
<td>$D \rightarrow 2$ $\rightarrow 3$ $\rightarrow 1$ $\rightarrow 6$ $\rightarrow 4$ $\rightarrow 5$</td>
<td>$D \rightarrow 1$ $\rightarrow 2$ $\rightarrow 3$ $\rightarrow 4$ $\rightarrow 5$</td>
</tr>
</tbody>
</table>

**Fig. 5: The decision tree**

**Model Behavior:**

The behavior of the algorithm was analyzed, using 10 problems with symmetric matrices whose elements were generated randomly through a subroutine RANDU, called by the main program. The generated numbers range from 1 to 99. Demands are random numbers and matrices of dimensions ranging from $7 \times 7$ to $15 \times 15$, with 5 of the problems using 2 vehicles and 5 of the problems using 3 vehicles for delivery, all of same capacity and equal to 60 units. The number of iterations of each
problem, for the final solution, ranging from 17 to 1161.

A model application with actual data, in a gas distribution company, based in the city of Santa Maria, Rio Grande do Sul, Brazil and which acts as a deposit, transporting to several cities in the interior of the State. The cities in which must be carried out deliveries, were grouped into two regions, determining two symmetrical problems of dimensions 16 x 16 and 14 x 14. In problem 1, 16 x 16 matrix, 2 vehicles were used, because the sum of the demands and the capabilities of vehicles. The final solution was obtained with 126 iterations. The problem 2, who used 3 vehicles, obtained the final solution with 3410 iterations. It is observed that the difficulty of obtaining the final solution grows with the number of vehicles used and the number of iterations depends on the dimension of the array, the homogeneity of its elements and the relationship between the sum of the demands and the sum of the capacities of vehicles used. The problems that use larger number of vehicles present greater difficulty to obtain the final solution, and therefore requires more computation time.

**Conclusion:**

The branching algorithm and limits developed in this work is in General and can be used with symmetrical and non-symmetrical problems traveling salesman simple, multiple traveling salesman and in transport routes problems with vehicle capacity constraints, obtaining always a great solution. It is observed that the difficulty of obtaining the final solution grows with the number of vehicles used and the number of iterations depends on the dimension of the array, the homogeneity of its elements and the relationship between the sum of the demands and the sum of the capacities of vehicles used. The problems that use larger number of vehicles present greater difficulty to obtain the final solution, and therefore requires more computation time.

**REFERENCES**


Sortica, G.B., 1982. Um modelo de Ramificações e Limites para Problemas de Rotas de Transporte. Dissertação de Mestrado. PPGEP, UFSM, Santa Maria, RS.