

Jan Fabry *

INSERTION METHOD FOR MULTIPLE MESSENGER PROBLEM WITH MULTIPLE DEPOTS

Messenger problem, as the variation of pickup and delivery problem, deals with the transport of a set of packages from their origins to given destinations. In reality, several vehicles have to be used to be able to satisfy all requirements within given time limit. Messengers can be located in one or multiple depots. Because of NP-hardness of the problem it is impossible, for most real problems, to find the optimal solution in acceptable time. Therefore, heuristic algorithms must be used. In the paper, insertion method for routes generation is presented. As the computational experiments in VBA for Excel show, obtained results can be improved using the exchange algorithm.

Keywords: Multiple messenger problem, insertion method, exchange algorithm.

1. Introduction

Messenger problem, as a variation of a pickup and delivery problem, is analogous to dial-a-ride problem (DARP), which Cordeau et al. [1] classify as the "one-to-one" problem. For transport of each person, a pickup location and a drop off location are given. Cordeau and Laporte [2] present models and algorithms for solving DARP, de Paepe et al. [3] investigate the computational complexity of this class of problems. Static DARP with multiple vehicles prepared in one depot for transport of persons is solved by Cordeau [4], which describes branch-and-cut method. For this type of problem with time windows, Toth and Vigo [5] used parallel insertion method, Jorgensen et al. [6] applied genetic metaheuristics. For packages transport, we prefer to use the designation messenger problem. In [7] heuristic algorithms for the multiple messenger problem located in one depot are developed as the modifications of nearest neighbour method, insertion algorithm and exchange method.

In the paper, the messenger problem with multiple vehicles located in multiple depots is solved. Multiple vehicles are going to be used especially in situations with the limited vehicle capacities in terms of shipment sizes, or in case that time limitation for routes duration are given. Due to the computational complexity of these problems, heuristic algorithms are developed for real instances. Modification of insertion method is used for generation of multiple routes and exchange algorithm is applied for their improvement.

2. Insertion method for multiple messenger problem with multiple depots

In messenger problem, a location of pickup and a location of delivery are given for each shipment. Suppose n packages have to be transported by messengers, all are known in advance. Let us denote all pickup nodes in the network by the even numbers and all delivery nodes by the odd numbers. In addition, if the package is picked up at node i it will be delivered at node $i + 1$. For service, K messengers in separated depots are available. Let us assume, because of the proposed algorithms, K is an odd number. In case the real number of messengers is even, a dummy messenger is introduced. Nodes $k = 1, 2, \dots, K$ specify depots, nodes $K + 1, K + 3, \dots, K + 2n - 1$ correspond to pickup locations and $K + 2, K + 4, \dots, K + 2n$ denote delivery locations. Minimal distances c_{ij} and associated travel times d_{ij} for all pairs of locations are given ($i, j = 1, 2, \dots, K + 2n; i \neq j$). In addition, T_{max} is defined as the acceptable time for each route duration. In the following algorithm we suppose that all vehicles will be used for shipments delivery, i.e. K routes will be generated. Let U_k^* be a sequence of h_k nodes included in route k of length z_k and duration T_k , and U_k be a set of nodes that can be still included in route k (in the certain step of routes generation). Number $u_i^k \in U_k^*$ corresponds to the node on position i in sequence U_k^* . Value of z is the total length of all generated routes.

The algorithm is based on inserting locations to current routes which extends their length. If insertion method is used for travelling salesperson problem, the route is obviously initialized

* Jan Fabry

Department of Econometrics, University of Economics Prague, Czech Republic
E-mail: fabry@vse.cz

by finding the farthest location s from the depot 1. Then, the cycle $[1, s, 1]$ is created as the initial route. Another location is inserted to the cycle with the objective to minimize its extension. This approach is repeated until all locations are incorporated in the Hamiltonian cycle. In case of multiple messenger problem, two important rules have to be respected: pickup location and related delivery location have to be inserted to the same route, and pickup location has to be inserted to the route in front of related delivery location. In addition, delivery location can be inserted exactly behind related pick-up one or behind several locations farther [8]. If multiple messengers are considered in the problem, finding the farthest location to each depot in the initialization step can be disadvantageous because that location can be very close to another depot. In such situation it would be inefficient to include the location in the route of the concerned messenger. In the proposed algorithm this weakness is eliminated. For each messenger k , we are searching for the location that is closer to depot k than to all other depots. If more locations satisfy this condition, the farthest one to depot k determines the initial route (see variables $LowDist$ and $LowLoc$). If no such location exists, i.e. all locations are farther to depot k than to any of remaining depots, as the initial location, it is taken the one with the minimal difference between its distance from depot k and minimal distance from other depots (see variables $UpDif$ and $UpLoc$). Because at the beginning each messenger can visit all locations, all sets U_k include all pickup and delivery nodes.

Step 1

For $k = 1, 2, \dots, K$ set $U_k = \{K + 2, K + 4, \dots, K + 2n\}$.

Step 2

For $k = 1, 2, \dots, K$ repeat

$LowDist = -\infty$; $UpDif = +\infty$;

for $\forall i \in U_k$ repeat

if $c_{ki} \leq \min_{\substack{l=1,2,\dots,K \\ l \neq k}} c_{li}$ then

if $c_{ki} > LowDist$ then

$LowDist = c_{ki}$; $LowLoc = i$;

else if $\left(c_{ki} - \min_{\substack{l=1,2,\dots,K \\ l \neq k}} c_{li}\right) < UpDif$ then

$UpDif = \left(c_{ki} - \min_{\substack{l=1,2,\dots,K \\ l \neq k}} c_{li}\right)$; $UpLoc = i$;

if $LowDist > -\infty$ then $s = LowLoc$ else $s = UpLoc$;

if s is even then

$U_k^* = \{k, s, s+1, k\}$; $z_k = c_{ks} + c_{s,s+1} + c_{s+1,k}$;
 $T_k = t_{ks} + t_{s,s+1} + t_{s+1,k}$;

else

$U_k^* = \{k, s-1, s, k\}$; $z_k = c_{k,s-1} + c_{s-1,s} + c_{s,k}$;
 $T_k = t_{k,s-1} + t_{s-1,s} + t_{s,k}$;

$h_k = 4$;

for $k = 1, 2, \dots, K$ repeat

$U_l = U_l - \{s\}$;

if s is even, then $U_l = U_l - \{s+1\}$ else $U_l = U_l - \{s-1\}$.

Step 3

$\Delta z = +\infty$;

For $k = 1, 2, \dots, K$ repeat

for all even $r \in U_k$ repeat

$\Delta z_{Imm_{rj}} = \min_{i=1,2,\dots,h_k-1} (c_{u_i^k r} + c_{r,r+1} + c_{r+1,u_{i+1}^k} + c_{u_i^k u_{i+1}^k})$;

$\Delta z_{Imm} = t_{u_j^k r} + t_{r,r+1} + t_{r+1,u_{j+1}^k} - t_{u_j^k u_{j+1}^k}$;

$\Delta z_{Lat_{rvw}} = \min_{\substack{i=1,2,\dots,h_k-2 \\ m=i+1,i+2,\dots,h_k-1}} (c_{u_i^k r} + c_{ru_{i+1}^k} + c_{u_i^k u_{i+1}^k} + c_{u_m^k r+1} + c_{r+1,u_{m+1}^k} + c_{u_m^k u_{m+1}^k})$;

$\Delta t_{Lat} = t_{u_i^k r} + t_{ru_{i+1}^k} - t_{u_i^k u_{i+1}^k} + t_{u_m^k r+1} + t_{r+1,u_{m+1}^k} - t_{u_m^k u_{m+1}^k}$;

if $T_k + \Delta t_{Imm} < Tmax$ and $\Delta z_{Imm_{rj}} < \Delta z$ then

$Imm = TRUE$; $Msg = k$; $Pick = r$; $After1 = j$;

$\Delta z = \Delta z_{Imm_{rj}}$; $\Delta T = \Delta t_{Imm}$;

if $T_k + \Delta t_{Lat} < Tmax$ and $\Delta z_{Lat_{rvw}} < \Delta z$ then

$Imm = FALSE$; $Msg = k$; $Pick = r$; $After1 = v$; $After2 = w$;

$\Delta z = \Delta z_{Lat_{rvw}}$; $\Delta T = \Delta t_{Lat}$;

if $Imm = TRUE$ and $\Delta z = +\infty$ then

insert nodes $Pick$ and $Pick + 1$ to sequence U_{Msg}^* to positions $After1 + 1$ and $After1 + 2$;

else if $\Delta z = +\infty$ then

insert nodes $Pick$ and $Pick + 1$ to sequence U_{Msg}^* to positions $After1 + 1$ and $After2 + 1$;

if $\Delta z = +\infty$ then

$h_{Msg} = h_{Msg} + 2$; $z_{Msg} = z_{Msg} + \Delta z$; $T_{Msg} = T_{Msg} + \Delta T$;

for $k = 1, 2, \dots, K$ repeat

$U_k = U_k - \{Pick\}$; $U_k = U_k - \{Pick + 1\}$.

Step 4

If $U_k = \emptyset$ for $\forall k = 1, 2, \dots, K$ then go to Step 5 else go to Step 3.

Step 5

$z = \sum_{k=1}^K z_k$.

End.

In each iteration, the related pickup and delivery locations are inserted in two possible ways [8]: the delivery location will be visited immediately after visiting the pickup location, or it will be visited later, after visiting other locations. In the first case, value $\Delta z_{Imm_{rj}}$ expresses the minimal extension of the current route of messenger k , if pickup node r and delivery node $r+1$ are inserted together behind the node on position j in sequence U_k^* . Value Δt_{Imm} is calculated to check whether it is acceptable to include those locations in the route due to time limitation.

Similarly, in the second case, value $\Delta zLat_{rvw}$ corresponds to the minimal extension of the route, if nodes r and $r + 1$ are inserted to sequence U_k^* separately: node r on position v and node $r + 1$ on position w . Variable $\Delta tLat$ is introduced to check feasibility in terms of time limitation.

After finding all acceptable insertions minimizing the extensions of individual routes, the most advantageous insertion of nodes $Pick$ and $Pick + 1$ is realized in the route of messenger Msg (in sequence U_{Msg}^*) behind position $After1$ (if $Imm = TRUE$) or behind positions $After1$ and $After2$ (if $Imm = FALSE$). All sets U_k are modified finally, i.e. nodes $Pick$ and $Pick + 1$ are excluded from all sets. The algorithm does not consider the situation that no messenger can accept the shipment due to time limitation. Making the proper corrections of the algorithm depends on real strategy of real distribution company.

3. Exchange algorithm

The application of insertion algorithm leads to the generation of multiple routes that can be improved using another heuristic approach. For the problem with multiple messengers starting in one depot, exchange method was proposed in [7]. In the paper, the method is modified for the problem considering multiple depots. Let us assume K generated routes given by sequences U_k^* ($k = 1, 2, \dots, K$). Node u_i^k is a member of related sequence U_k^* on position i .

Step 1

For each route the shipment is found, i.e. the pair of pickup and delivery locations, whose exclusion from the route brings the maximal reduction of the length. This shipment is the candidate for the possible exchange. Let us denote this reduction $\Delta zExcI_k$ ($k = 1, 2, \dots, K$). The maximal reduction is selected:

$$\Delta zExcI_m = \max_{k=1,2,\dots,K} \Delta zExcI_k.$$

Let us assume this reduction in route m corresponds to nodes r and $r + 1$ on positions v and w in sequence U_m^* .

Step 2

For all routes $k = 1, 2, \dots, K$, $k \neq m$ the most advantageous insertion of nodes r and $r + 1$ is found using the approach described in Step 3 of insertion method. For each route its minimal extension $\Delta zExt_k$ ($k = 1, 2, \dots, K$, $k \neq m$) is calculated. Their minimal value is selected:

$$\Delta zExt_p = \min_{\substack{k=1,2,\dots,K \\ k \neq m}} \Delta zExt_k.$$

Let this extension of route p correspond to the insertion of node r behind node u_i^p and node $r + 1$ behind node u_j^p .

Step 3

If $\Delta zExcI_m = \Delta zExt_p$ then the exchange is realized, i.e. routes m and p are modified, and it is continued with Step 1, else with Step 4.

Step 4

End.

It should be emphasized that the proposed algorithm does not exchange shipments but transfers them from one route to another. Because in Step 2 time limitation $Tmax$ has to be respected, real exchange could be executed in case of infeasibility. However, it is necessary to check feasibility of both routes m and p . The algorithm can be modified: all pairs of shipments across all routes are tested to feasibility and optimality.

If the algorithm finishes because the transfer cannot be realized (in Step 3), we can continue with Step 1 so that route m is added into "tabu list" and will be ignored when finding maximal reduction $\Delta zExcI_m$. Of course, it is possible to add other routes into "tabu list" if necessary.

4. Computational experiments

In [9] the application in VBA for Excel was developed on the base of proposed method and computational experiments were executed. They related the real matrix of minimal distances between 650 locations in the Czech Republic. Instances with 10, 20, 30 and 40 shipments were generated randomly for 3, 5, 7 and 11 messengers. In each category, 10 samples were examined (see Table 1).

Results of experiments

Table 1

Depots	Shipments	IN x NN	NN Ex	Dist NN	IN Ex	Dist IN	IN Ex x NN Ex
		%	%	%	%	%	%
3	10	90.0	80.0	2.1	30.0	0.5	70.0
5	20	60.0	60.0	1.3	10.0	0.2	50.0
7	30	80.0	50.0	1.2	20.0	0.3	70.0
11	40	60.0	70.0	1.8	10.0	0.2	60.0
		72.5	65.0	1.6	17.5	0.3	62.5

Besides insertion method, also nearest neighbour algorithm was applied to generated data. In 72.5 % of all instances insertion method provided better results than nearest neighbour algorithm (see column IN x NN in Table 1). Using exchange algorithm, 65 % of routes generated by nearest neighbour algorithm were improved (NN Ex), while in case of insertion method the value was 17.5 % (IN Ex). Even after the improvement, 62.5 % of solutions obtained by insertion approach were better (IN Ex x NN Ex). The application of exchange algorithm leads to shortening routes by 1.6 % of the total length in case of nearest neighbour algorithm (Dist NN) and 0.3 % in case of insertion method (Dist IN).

5. Conclusion

Results of computational experiments indicate higher efficiency of proposed insertion method than nearest neighbour algorithm described in [9]. The application of exchange algorithm significantly improved solutions obtained by nearest neighbour algorithm. Although in case of insertion method the improvement

is not so expressive, final results show its higher efficiency in comparison with nearest neighbour algorithm. The statistically significant generalization of interesting conclusions requires the stronger analysis and wider experiments on real data. Greater savings would be probably achieved by the modifications mentioned above. Similarly to other special vehicle routing problems [10], time windows can be considered in multiple messenger problem. The proposed heuristic methods can be also extended to multiple messenger problems with a limited capacity of vehicles. As it was proved on the single vehicle messenger problem [8], insertion algorithm has its significance especially in case of dynamic problems. Future research will be aimed at using heuristic algorithms just in multiple messenger problems with multiple depots.

Acknowledgements

This research was supported by the project F4/11/2013 funded by the Internal Grant Agency of the University of Economics Prague.

References

- [1] CORDEAU, J.-F., LAPORTE, G., ROPKE, S.: Recent Models and Algorithms for One-to-One Pickup and Delivery Problems, pp. 327-357. In: GOLDEN, B. L., RAGHAVAN, S., WASIL, E. A. (eds.): *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, 2008.
- [2] CORDEAU, J.-F., LAPORTE, G. A.: The Dial-a-Ride Problem: Models and Algorithms. *Annals of Operations Research*, vol. 153, pp. 29-46, 2007.
- [3] DE PAEPE, W. E., LENSTRA, J. K., SGALL, J., SITTERS, J. A., STOUGIE, L.: Computer-Aided Complexity Classification of Dial-a-Ride Problems, *INFORMS J. on Computing*, vol. 16, pp. 120-132, 2004.
- [4] CORDEAU, J.-F.: A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Operations Research*, vol. 54, pp. 573-586, 2006.
- [5] TOTH, P., VIGO, D.: Heuristic Algorithms for the Handicapped Persons Transportation Problem. *Transportation Science*, vol. 31, pp. 60-71, 1997.
- [6] JORGENSEN, R. M., LARSEN, J., BERGVINSDOTTIR, K. B.: Solving the Dial-a-Ride Problem Using Genetic Algorithms. *J. of the Operational Research Society*, vol. 58, pp. 1321-1331, 2007.
- [7] FABRY, J., KOBZAREVA, M.: *Multiple Messenger Problem*. Proc. of Mathematical Methods in Economics '12, Karvina, pp. 141-147, 2012.
- [8] FABRY, J.: Dynamic Messenger Problem. *Communications - Scientific Letters of the University of Zilina*, No. 4, pp. 66-69, 2007.
- [9] PRIBYLOVA, L.: *Heuristic Methods for Capacitated Messenger Problems (in Czech)*, Diploma thesis, University of Economics: Prague, 2014.
- [10] CICKOVA, Z., REIFF, M., SURMANOVA, K.: *Split Delivery Vehicle Routing Problem with Time Windows*. Mathematical Methods in Economics 2014, Olomouc, pp. 128-132, 2014.