Thai Journal of Mathematics : 205–216
Special Issue: Annual Meeting in Mathematics 2017
http://thaijmath.in.cmu.ac.th
ISSN 1686-0209

Multiple Depot Vehicle Routing Problems on Clustering Algorithms

Kanokon Leungsubthawee[†,‡], **Supalin Saranwong**[†,‡] **and Chulin Likasiri**[†,‡,1]

[†]Center of Excellence in Mathematics and Applied Mathematics
Department of Mathematics, Faculty of Science, Chiang Mai University
Chiang Mai 50200, Thailand
[‡]Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd.
Bangkok 10400, Thailand
e-mail : kanokon_leungs@cmu.ac.th (K. Leungsubthawee)
supalin_s@cmu.ac.th (S. Saranwong)
chulin.l@cmu.ac.th (C. Likasiri)

Abstract : This work addresses the clustering problem with two types of clients via the multiple depot vehicle routing problem (MDVRP). The objective function is to minimize the total distance traveled in the system. Two algorithms are proposed to tackle the difficulty of this problem. In the first algorithm, clients are randomly assigned to their closet depots while in the second algorithm each depot in the unassigned depot list is assigned sequentially to its nearest unassigned client. Comparisons of solutions obtained from the proposed algorithms and the optimal solutions show that in small size problems, the objective functions from the first algorithm are closer to the optimal solutions than those from the second algorithm. In larger problems, however, the second algorithm works better than the first because the difference between the number of clients and number of depots is increased. More feasible solutions can also be obtained from the second algorithm in all problems sizes. It can thus be seen that the ratio between number of clients and number of depots affects to the performance of the proposed algorithms.

Keywords : algorithm; cluster problem; multiple depot vehicle routing.
2010 Mathematics Subject Classification : 46N10; 90B06.

---

[1]Corresponding author.

1 Introduction

An important issue in logistics is the efficiency of vehicle routing management in minimizing the total transportation cost. Not only do system managers make decisions on the number of used vehicles, they also have to specify which customers are served by which vehicle and what sequence of customers is served by each vehicle. This problem is known as the vehicle routing problem (VRP). Heuristics are usually proposed to tackle its difficulty. Routing problem can also help solve other issues such as clustering problems with the objective to minimize the total travelling time or the distance traveled in the system, the latter of which is the main focus of this study.

A clustering problem involves grouping a set of data using distance or similarity measures among individual data. Clustering problems and algorithms have been proposed by many researchers. Gonzalez [1] proposed an approximation algorithm to solve the problem of clustering a set of points to minimize the maximum intercluster distance. The complexity of this algorithm is $O(nk)$ where n is the number of points and k is the number of clusters. Bradley et al. [2] proposed an approach for assigning points to clusters based on the concave minimization model, by formulating a mathematical model to minimize the total distance from each node to the nearest center and comparing it with other algorithms. Negreiros and Palhano [3], addressing the capacitated centered clustering problem (CCCP), found a set of clusters with limited capacity with minimum distance in each cluster, and proposed a heuristic algorithm for solving this problem. Later, Chaves and Lorena [4] proposed a metaheuristic algorithm for solving CCCP, the main idea of the algorithm being to identify the searching space by generating solutions and clustering them into groups that are subsequently explored with a local search heuristic.

A clustering problem can be solved optimally using the traveling salesman problem (TSP) as described in Lenstra and Kan [5] i.e. finding a cyclic permutation of cities such that the total distance between adjacent cities under the permutation is minimized. They transformed cities and the distance between the corresponding cities as points and the distance between two points, respectively, making the TSP route an optimal clustering of points with minimum total distance. Climer and Zhang [6] proposed an algorithm based on a variation of TSP for clustering problems. A TSP variation, the vehicle routing problem (VRP), deals with how we design and manage customer routing in order to minimize the total cost (transportation or distance-based cost). Each route begins at the depot or center, serves customers and ends back at the depot or center under the conditions that each customer belongs to only one route and the total demand of served customers cannot exceed the capacity of the serving vehicle. In VRP, the system may involve multiple routes so as to minimize the total distance traveled. Assigning customers to routes in VRP can be considered a clustering problem. There are several VRPs depending on the types of service and limitations. Applications on real world problems include a VRP model in the collection of olive oil in Tunisia, proposed by Lahyani et al.[7], and a waste collection problem described by Buhrkal et al. [8] who proposed an algorithm and applied it to a Danish company. One variant of VRP is multiple depot vehicle routing problem (MDVRP), which involves finding routes with multiple depots.

There are 2 ways to solve VRP: Exact methods and heuristics. The exact methods give the optimal solution but involve considerable execution time, so they are only suitable for small size problems. Examples of the exact methods are branch and bound, branch and cut and cutting plane methods. Christofides et al. [9] proposed a branch and bound method for solving VRP and compared the computational results with those from other methods. Laporte et al. [10] proposed a class of multi-depot vehicle routing and location routing problems, which they transformed into graph and then solved using a branch and bound algorithm. Baldacci and Mingozzi [11] proposed an exact method for solving an extended model of VRP, using LP-relaxation and Lagrangean relaxation to derive the lower bounds. Even though heuristics may not yield optimal solutions, they are a popular method for solving NP-hard problems because their execution time is measurably shorter than that of the exact method. Many approximation algorithms also belong to this class, including genetic algorithm, ant colony optimization and tabu search, among others. Crevier et al. [12] studied a variant of MDVRP and proposed a heuristic for solving an extension of MDVRP where depots can act as intermediate replenishment facilities along a route. The proposed algorithm combined the adaptive memory principle and tabu search algorithm. Mancini [13] proposed a mathematical formulation to minimize the total cost of delivery operations over a fixed time-horizon. An Adaptive Large Neighborhood Search-based metaheuristic was proposed and computational results were illustrated in this work. Wang et al. [14] proposed a heuristic for solving the max-min split delivery multi-depot vehicle routing problem with minimum service time requirement. Their heuristic consists of 3 stages: initialization of a feasible solution without split service time and no minimum service time requirement; improvement of the solution by splitting service time but ignoring the minimum service time requirement via local search; and solving linear programing with the added constraints to ensure that minimum service time requirements are satisfied. Oliveira et al. [15] proposed a cooperative coevolutionary algorithm to minimize the total cost of the MDVRP. They decomposed the problem into subproblems, then solved subproblems and simultaneously evolved solutions to the main problem.

In this work, we propose algorithms for assigning clients to clusters, where each cluster consists of one depot and two types of clients, the objective being to minimize the total distance traveled. The problem can be formulated as a multiple depot vehicle routing problem serving two types of clients (suppliers and customers) on each route. Currently available algorithms and heuristics are constructed to solve MDVRP with only one type of client, and so they cannot be used to solve the problems being studied. The remainder of this paper is organized as follows: The mathematical formulation is described in Section 2; the proposed algorithms and operation counts are stated in Section 3; simulation results are discussed in Section 4 and finally, the conclusions are given in Section 5.

2 Mathematical Formulation

This section presents the mathematical programming of multiple depot capacitated vehicle routing problems (MDVRP). The MDVRP is defined by a graph $G = (V, A)$ where $V = \{1, 2, ..., p + m + n\}$ denotes the set of nodes, and $A = \{(i, j)|i, j \in V\}$ denotes the set of edges between nodes i and j. The routing denoted as $k = 1, 2, ..., p$ starts and ends at node $i = 1, 2, .., p$ which are the depots with Q capacity and a fixed cost for each depot as f_i. The suppliers are indexed as $i = p + 1, p + 2, ..., p + m$ with a supply for each supplier as s_i. The clients are indexed as $i = p + m + 1, p + m + 2, ..., p + m + n$ and the demand of each client as d_i. The transportation cost from node i to node j is denoted as c_{ij} and can be either the distance traveled or time spent. Decision variables are:

$$x_{ij}^k = \begin{cases} 1 & \text{, if edge } (i,j) \text{ is in cluster } k, \\ 0 & \text{, otherwise} \end{cases}$$

where $i, j \in \{1, 2, ..., p + m + n\}, k \in \{1, 2, ..., p\}$.

$$y_j^k = \begin{cases} 1 & \text{, if depot } j \text{ is in cluster } k, \\ 0 & \text{, otherwise} \end{cases}$$

where $j \in \{1, ..., p\}, k \in \{1, 2, ..., p\}$.

$$\begin{aligned} \text{Let } X^k &= [x_{1j}^k \; x_{2j}^k \; ... \; x_{(p+m+n)j}^k]^T, j \in \{1, 2, ..., p + m + n\} \\ Y^k &= [y_j^k]^T, j \in \{1, 2, ..., p\} \\ U^k &= [u_j^k]^T, j \in \{p + 1, p + 2, ..., p + m + n\} \\ C &= [c_{1j} \; c_{2j} \; ... \; c_{(p+m+n)j}], j \in \{1, 2, ..., p + m + n\} \text{ and} \\ F &= [f_j], j \in \{1, 2, ..., p\} \end{aligned}$$

$$\min \quad \sum_{k=1}^{p} CX^k + \sum_{k=1}^{p} FY^k \tag{2.1}$$

$$\text{s.t.} \quad A_1X^k - B_1Y^k = 0 \tag{2.2}$$

$$\sum_{k=1}^{p} A_2X^k = 1 \tag{2.3}$$

$$B_2Y^k = 1 \tag{2.4}$$

$$A_3X^k = 0 \tag{2.5}$$

$$\sum_{k=1}^{p} A_4X^k = 0 \tag{2.6}$$

$$\begin{aligned}
\sum_{k=1}^{p} A_4 X^k & & &= 0 && (2.7)\\
A_5 X^k & & &\leq D_1 && (2.8)\\
A_6 X^k & &+ E_1 U^k &\leq D_2 && (2.9)\\
& D_3 \leq & E_2 U^k &\leq D_4 && (2.10)\\
& & X^k \text{ and } Y^k &\text{ are binary} &&\\
& & U^k &\geq 0 &&
\end{aligned}$$

Objective function (2.1) is to minimize the total cost including transportation cost and fixed cost. Constraints (2.2) ensure that each routing starts from the depot and continues to supplier (customer) until it ends back at the depot where A_1 is the $6p \times (p+m+n)^2$ matrix whose elements are 0 or 1 and $B_1 = [2I_p\ I_p\ I_p\ I_p\ I_p\ 2I_p]^T$. Constraint (2.3) implies that each supplier (customer) can be on only one route where A_2 is the $2(m+n) \times (p+m+n)^2$ matrix whose elements are 0 or 1. Constraints (2.4) ensure that each depot is on only one route where $B_2 = [e\ 0_{p\times 1}\ I_p]$. Constraints (2.5) ensure that the entering arc to each supplier (customer) and the leaving arc from this supplier (customer) is on the same route where A_3 is the $(p+m+n) \times (p+m+n)^2$ matrix whose elements are 0 or 1. Constraint (2.6) ensures that there are no route between the different clients i.e., each supplier does not connect with customer where A_4 is the $2 \times (p+m+n)^2$ matrix whose elements are 0 or 1. Constraint (2.7) imply that the total supplies (demands) of the suppliers (customers) on any one route do not exceed the capacity of the depot serving that route where $A_5 = \begin{bmatrix} s_j \\ d_j - s_j \end{bmatrix}, j \in \{1, 2, ..., p+m+n\}$ is the $2p \times (p+m+n)^2$ matrix, and $D_1 = [Q\ Q\ ...\ Q]^T$ is the $2p \times 1$ matrix. Constraints (2.8) ensure that there will be no cycle in each route where A_6 is a constant matrix with dimension $2(m+n)^2 \times (p+m+n)^2$, E_1 is the $[2(m+n)^2 + 2p] \times (m+n)$ matrix whose elements are $1, 0, -1$, and $D_2 = [Q - d_j\quad Q - s_j]^T, j \in \{p, p+1, ..., p+m+n\}$. Constraints (2.9) imply that value of U^k is between supply (demand) and capacity of depot where $E_2 = [I_{m+n}\ \ I_{m+n}\ \ ...\ \ I_{m+n}]^T, D_3 = [d_{p+1}\ \ d_{p+2}\ \ ...\ \ d_{p+m+n}\ \ s_{p+1}\ \ s_{p+2}\ \ ...\ \ s_{p+m+n}]^T$, and $D_4 = [Q\ Q\ ...\ Q]^T$. This model is to assign suppliers and customers to clusters, each having one depot or center as shown in Figure 1.

3 The Proposed Algorithms and Operation Counts

The multiple depot vehicle routing problem is difficult to solve because it is an NP-hard problem. The two algorithms proposed can solve this problem in a reasonable processing time. The proposed algorithms assign suppliers (customers) to a depot according to the following rules:

1. The supply of each supplier cannot exceed the capacity of the depot;

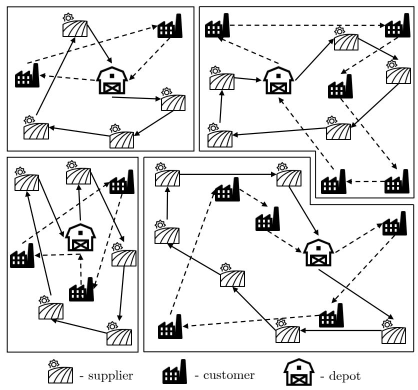

Figure 1: Example of 4 clusters with 20 suppliers and 15 customers.

2. The distance between a supplier and a depot is in a fixed radius;
3. A supplier is assigned to a cluster by connecting to an end-point node whose distance to the supplier is minimum.

To guarantee feasibility, the fixed radius is assumed to be the minimum distance from each depot to the farthest supplier (customer).

3.1 Algorithm A: Minimum Distance

In this algorithm, the distances from suppliers to depots are sorted and then the supplier with the minimum distance is assigned to the depot nearest to them (within the given radius) provided that their supply does not exceed that depot's capacity. Repeat the processes until all suppliers have been assigned.

Algorithm A is described in detail only in the supplier-depot part as follows:

For each unassigned supplier i,

If the supply does not exceed the remaining capacity of some depot,

If the distance between supplier i and its nearest depot is in a fixed radius,
assign supplier i to that depot, and
uupdate the remaining capacity of the chosen depot.

Else, supplier i cannot be assigned to any depot within the given radius, the problem is infeasible in this radius.

End.

Else, supplier i cannot be assigned to any depot, the problem is infeasible.

End.

Repeat the process until the unassigned supplier set is empty.

End.

The operation count of Algorithm A is $mp + np$. If m is a very large number compared to n i.e. $m \gg n$, then the complexity of this algorithm is $O(mp)$.

3.2 Algorithm B: Minimum Distance with Balancing Depot

In Algorithm B, the supplier located nearest to the first depot is chosen and assigned to the first depot, and then each of the remaining depots is sequentially assigned to its closest unassigned supplier (within the given radius) provided that their supply does not exceed the capacity of that depot.

This algorithm can be written in detail only in the supplier-depot part as follows:

Do while the unassigned supplier set is not empty,

While the unassigned depot list is not empty,

If the nearest suppliers supply i does not exceed the remaining capacity of depot j,

If the distance between supplier i and depot j is in the fixed radius,
assign the supplier i to depot j,
update the remaining capacity of depot j, and
move depot j to the end of the unassigned depot list.

Else, remove depot j from the list of unassigned depots and move on to the next depot.

End.

Else,

remove depot j from the list of unassigned depots.

If the unassigned depot list is empty,
the problem is infeasible.

Else, move to the next depot.

End.

End.

Move to the next unassigned supplier.

End.

End.

The operation count of Algorithm B is $m^2 + m + n^2 + n$. If m is a very large number compared to n i.e. $m \gg n$, then the complexity of this algorithm is $O(m^2)$.

The processes of the 2 proposed algorithms are shown in Figure 2.

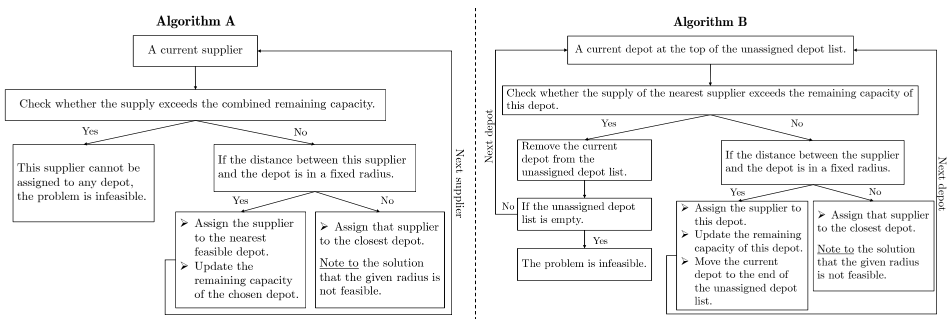

Figure 2: The processes of the proposed algorithms.

4 Simulation Results

We formulate our mathematical model on AIMMS where the models are solved by CPLEX solver. The multiple depot vehicle routing problem is NP-hard. However, an optimal solution can be found when the problem size is small. Small size optimal solutions obtained from CPLEX are compared with the proposed algorithms. The problem sizes constructed are $10 \times 3 \times 3$ (10 suppliers $\times$ 3 depots $\times$ 3 customers), $10 \times 3 \times 5$, $10 \times 3 \times 7$, $10 \times 3 \times 9$, and $10 \times 3 \times 11$. In each size, 30 problems are solved on Intel® Xeon ® CPU@2.30 GHz 2.29 GHz (2 processors) with 64 GB of RAM. The random samplings, distances between suppliers, customers and depots are normally generated with $\mu = 150$ and $\sigma = 40$. The combined supplies of suppliers is at 60% of the combined capacity of depots and the combined demands of customers is at 60% of the combined supplies of suppliers. The supplies (demands) of the suppliers (customers) are generated from the normal distribution with the different μ depending on the sizes of the problems and σ is 30% of μ. The solutions obtained via the proposed algorithms along with their corresponding processing times are compared with those of the optimal solution from CPLEX.

Table 1: Results of data simulations for small size problems

size	Optimal		Algorithm A			Algorithm B		
	Execution times (s)	Total distance	# of feasible solutions	Execution times (s)	Total distance	# of feasible solutions	Execution times (s)	Total distance
$10 \times 3 \times 3$	121.98	1,089.77	29	0.039	1,394.52	30	0.087	1,461.47
$10 \times 3 \times 5$	1,339.90	1,224.63	27	0.054	1,617.15	30	0.084	1,678.10
$10 \times 3 \times 7$	2,160.61	1,321.50	30	0.050	1,745.77	30	0.038	1,811.57
$10 \times 3 \times 9$	2,525.91	1,422.67	29	0.051	1,927.28	30	0.040	1,935.50
$10 \times 3 \times 11$	12,128.38	1,575.27	25	0.057	2,068.92	29	0.088	2,013.83

As shown in Table 1, although CPLEX gives the minimum total distance, its processing time increases considerably when the problem size increases even slightly. When a comparison is made between the 2 algorithms, the total distance yielded by Algorithm A is shown to be shorter than that from Algorithm B except in problem size 10x3x11. The optimality gaps of Algorithm A are in $21 - 24\%$ while those of Algorithm B are in $21 - 27\%$ The numbers of feasible solutions yielded by Algorithm B are $3 - 14\%$ higher than those from Algorithm A.

In larger problems, the distances between a pair of nodes are normally generated with $\mu = 150$ and $\sigma = 40$. Suppliers combined supply is at 80% of the combined capacity of depots; customers combined demand is at 80% of suppliers combined supply. The supplies (demands) of the suppliers (customers) are generated from the normal distribution with the difference μ depending on the sizes of the problems and σ being 30% of μ. The problems are simulated and solved via the 2 proposed algorithms on Intel® Xeon® CPU E5-2667 v4 @ 3.20 GHz 3.20 GHz (2 processors) with 64 GB of RAM. In each case, 30 problems are generated.

Table 2: Results of simulations for larger size problems.

size	Algorithm A		Algorithm B	
	Execution times (s)	Total distance	Execution times (s)	Total distance
$250 \times 25 \times 250$	0.09	39,738.47	0.48	32,950.83
$250 \times 50 \times 250$	0.10	43,067.53	0.45	40,569.57
$250 \times 100 \times 250$	0.12	51,968.50	0.47	55,894.97
$500 \times 25 \times 500$	0.17	69,923.43	1.82	49,182.67
$500 \times 50 \times 500$	0.20	68,878.80	1.83	56,497.07
$500 \times 100 \times 500$	0.26	75,853.47	1.79	71,653.90
$1000 \times 25 \times 1000$	0.37	130,547.80	7.21	74,112.00
$1000 \times 50 \times 1000$	0.40	119,170.17	7.03	80,993.80
$1000 \times 100 \times 1000$	0.53	118,202.40	7.38	95,726.33

As shown in Table 2, the processing times required by Algorithm A are $75 - 95\%$ shorter than those required by Algorithm B, while the total distances obtained via Algorithm B are 5-45% shorter than those from Algorithm A except in problem size $250 \times 100 \times 250$. It can be observed that among the problems with the same number of clients, the differences between objective function values obtained from Algorithm A and B decrease when the number of depots increase. In all instances, both algorithms can result in feasible solutions. Therefore, the columns indicating the number of feasible solutions are omitted.

From Tables 1 and 2, it can be concluded that the ratio between number of clients and number of depots affects the performance of the proposed algorithms. Algorithm A works better than Algorithm B when the ratio between number of clients and number of depots is less than 3 (problem sizes $10 \times 3 \times 3, 10 \times 3 \times 5, 10 \times 3 \times 7, 10 \times 3 \times 9$, and $250 \times 100 \times 250$). When the ratios between number of clients and number of depots are over 3, Algorithm B results in a better solution. This is due to the fact that the searching space in each iteration of Algorithm A depends on the number of depots while the searching space of Algorithm B depends on the number of clients.

5 Conclusions

In this work, we create two algorithms based on MDVRP to solve clustering problems. The procedure assigns clients to clusters with the objective of minimizing the total distance in the system. The proposed algorithms are able to solve larger size problems within a reasonable time. Simulation data are generated from normal distribution. Since the real data are often large in number and right-skewed normally distributed, the larger data set used to test the proposed algorithms are randomly generated with a normal distribution having a mean depending on the size of the problem and SD of 30% of mean. In small size problems, the solutions obtained from the proposed algorithms are slightly different, with Algorithm A working better than Algorithm B in most cases. In small size problems, the number of feasible solutions obtained from Algorithm B is 13.79% greater than that from Algorithm A. In larger problems, however, Algorithm B generally yields better solutions. The objective functions obtained from Algorithm B are approximately 18.09% better than those obtained from Algorithm A. Both algorithms result in a feasible solution in all larger size problems. As for processing time, obtaining an exact solution from CPLEX for the problem size with 24 nodes takes more than 3 hours while the two proposed algorithms can solve large size problems in less than 8 seconds.

The assignment strategies of the two algorithms show different viewpoints. In Algorithm A, a random customer is assigned to the depot nearest to them while in Algorithm B, the ordered depot selects the nearest client to be served. Note that when the numbers of clients and depots are large and the former is greater than the latter, Algorithm B yields a better solution because the searching space in each iteration of B depends on the number of clients. When the number of clients is slightly greater than the number of depots, however, the searching space of Algorithm A is close to that of Algorithm B, but Algorithm A works better. This is because Algorithm B tries to balance the number of assigned clients in each depot by removing the assigned depot to the end of the list while in Algorithm A, clients can freely choose the nearest depot.

Acknowledgements : This research is supported by the Department of Mathematics, Faculty of Science, Chiang Mai University. Appreciation is extended to the Centre of Excellence in Mathematics, CHE for providing financial support and to Ms. Wiriya Sungkhaniyom for proofreading. The computers for testing algorithms in large size problems are devices of Faculty of Engineering, Chiang Mai University with the aids of Asst. Prof. Chaiy Rungsiyakull. The authors are grateful to the referees for their careful reading of the manuscript and their useful comments.

References

[1] T.F. Gonzalez, Clustering to minimize to maximum intercluser distance, Theoretical Computer Science 38 (1985) 293–306.

[2] P.S. Bradley, O.L. Mangasarian, W.N. Street, Clustering via Concave Minimization, Advances in Neural Information Processing Systems 19 (1997) 368–374.

[3] M. Negreiros, A. Palhano, The capacitated centred clustering problem, Computers and Operations Research 33 (2006) 1639–1663.

[4] A.A. Chaves, L.A.N. Lorena, Clustering search algorithm for the capacitated centered clustering problem, Computers and Operations Research 37 (2010) 552–558.

[5] J.K. Lenstra, A.H.G. Rinnooy Kan, Some simple applications of the travelling salesman problem, Operational Research Quarterly 26 (4) (1975) 717–733.

[6] S. Climer, W. Zhang, Rearrangement clustering: pitfalls, remadies, and applications, Journal of Machine Learning Research 7 (2006) 919–943.

[7] R. Lahyani, L.C. Coelho, M. Khemakhem, G. Laporte, F. Semet, A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia, Omega 51 (2015) 1–10.

[8] K. Buhrkal, A. Larsen, S. Ropke, The waste collection vehicle routing problem with time windows in a city logistics context, Procedia-Social and Behavioral Sciences 39 (2012) 241–254.

[9] N. Christofieds, A. Mingozzi, P. Toth, Exact algoritms for the vehicle routing problem, based on spanning tree and shortest path relaxations, Mathematical Programming 20 (1981) 255–282.

[10] G. Laporte, Y. Nobert, S. Taillefer, Solving a family of multi-depot vehicle routing and location-routing problems, Transportation Science 22 (3) (1988) 161–172.

[11] R. Baldacci, A. Mingozzi, A unified exact method for solving different classes of vehicle routing problems, Mathematical Programming 120 (2009) 347–380.

[12] B. Crevier, J.F. Cordeau, G. Laporte, The multi-depot vehicle routing problem with inter-depot routes, European Journal of Operational Research 176 (2007) 756–773.

[13] S. Mancini, A real-life multi depot multi period vehicle routing problem with a heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Metaheuristic, Transportation Research Part C 70 (2016) 100–112.

[14] X. Wang, B. Golden, E. Wasil, R. Zhang, The minmax split delivery multi-depot vehicle routing problem with minimum service time requirement, Computers and Operations Research 71 (2016) 110–126.

[15] F.B. Oliveira, R. Enayatifara, H.J. Sadaeia, F.G. Guimaresb, J. Potvin, A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem, Expert Systems With Applications 43 (2016) 117–130.

(Received 17 March 2017)
(Accepted 30 June 2017)