A reactive adaptive memory metaheuristic for solving HFFVRP

S.A. MirHassani* and Z. Saadati

Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, Tehran, Iran

(Received 20 November 2013; accepted 12 August 2014)

Product distribution is one of the most important problems in supply chain management. The heterogeneous fixed fleet vehicle routing problem (HFFVRP) is a variant of VRP that aims to determine the routes for a fleet of vehicles while satisfying demand and minimizing cost. The fleet composition is fixed and it consists of various types of vehicles. These vehicles differ according to their variable cost. In this paper, a reactive adaptive memory metaheuristic was developed which made use of the Tabu Search as an improvement procedure (ReAMTS) in order to solve the HFFVRP. The method was implemented and evaluated on two sets of test problems, which are used in the literature. In addition, its performance was compared with an exact method and a set of tractable instances. The computational results demonstrated high-quality results within a reasonable time.

Keywords: reactive adaptive memory; vehicle routing problem; heterogeneous fixed fleet; metaheuristic; Tabu search

1. Introduction

Distribution of goods is one of the major issues in the transportation industry and commercial companies that affect the final prices and customers’ satisfaction. A good distribution system can save resources of companies. VRP may be regarded as one of the successful stories of operation research (OR). After the first paper by Dantzig and Ramser (1959) on VRP, many variants and extensions have been introduced by other researchers. One of the well-known problems in this field goes back to the heterogeneous vehicle in which a problem involves a fleet with different types of vehicles starting from depot and serves a number of customers at different geographic locations and different demands. To the best of the present researchers’ knowledge, two variants of VRP with heterogeneous fleet (HFFRP) have been studied in the literature. In the first one, it was supposed that the number of vehicles of each type was unlimited and the aim was to find the optimal set of vehicles to be scheduled. This problem is called the fleet size and mixed VRP (FSMVRP) that is usually used for tactical decisions where the fleet is not yet purchased and selecting the number of vehicles has to be acquired. In the second type, there was a fixed fleet of vehicles, which meant the fleet was set and the number of vehicles of each type was fixed. This problem is called heterogeneous fixed fleet VRP (HFFVRP) that is usually used for operational decisions in which computing the trips and assigning them to vehicles are required. The FSMVRP may be considered as a particular HFFVRP where the number of vehicles of each type is equal

*Corresponding author. Email: a_mirhassani@aut.ac.ir

© 2014 The Author(s). Published by Taylor & Francis.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License http://creativecommons.org/licenses/by/3.0/, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The moral rights of the named author(s) have been asserted.
to the number of customers. For more details, the readers can refer to Hoff, Andersson, Christiansen, Hasle, and Lokketangen (2010).

In this work, the heterogeneous fixed fleet vehicle routing problem (HFFVRP) was investigated. As noted in Tarantilis, Kiranoudis, and Vassiliadis (2003) and according to the experience, solving the HFFVRP is more difficult than the classical VRP and FSMVRP. The most effective metaheuristic methods use local search procedures based on intra- or inter-route customer exchanges for solving the classical VRP or FSMVRP. For classic VRPs with identical vehicles, the feasibility check or evaluation of these moves is an easy task. On the other hand, the feasibility check or the evaluation of a move for the HFFVRP requires the reassignment of routes to the vehicles, which is a complicated task due to the limited (fixed) number of heterogeneous vehicles.

The HFFVRP is an NP-hard problem since it includes the VRP as a special case where the number of vehicles types is just one and the number of vehicles is unlimited. To the best of the researchers’ knowledge, no exact algorithms have been developed for HFFVRP. Taillard (1999) was the first to propose an algorithmic methodology for this problem. He developed a heuristic column generation method for solving the HFFVRP. In this method, for each vehicle type and unlimited number of vehicles, a homogeneous VRP is solved by employing the adaptive memory procedure (AMP) (Rochat & Taillard, 1995). A Tabu search algorithm is used to generate a set of good solutions in AMP. Then, single vehicle routes are extracted and are combined into a partial solution to act as an initial point for the next step. The process is repeated and routes are memorized as candidates for the final solution to the HFFVRP. The best solution to the HFFVRP is found by solving an integer linear program where each column corresponds to a candidate route.

Tarantilis et al. (2003) and Tarantilis, Kiranoudis, and Vassiliadis (2004) proposed a list-based threshold accepting algorithm (LBTA) and a backtracking adaptive threshold accepting algorithm (BATA) for solving the HFFVRP. Threshold accepting is a deterministic version of simulated annealing in which a threshold value $T$ is specified as the upper bound on the allowed quantity of objective function increase (uphill moves are admissible). In the LBTA, a list of values for $T$ is considered while the value of $T$, in the BATA, is allowed to increase through the search. Two algorithms of LBTA and BATA use two-opt moves, 1–1 exchanges and 1–0 exchanges while performing the local search.

Li, Golden, and Wasil (2007) adapted their record-to-record travel algorithm for the VRP (Li, Golden, & Wasil, 2005) to solve the HFFVRP and called it HRTR. Record-to-record travel, like the threshold accepting, as a deterministic version of simulated annealing was primarily proposed by Dueck and Scheuer (1990). Tarantilis and Kiranoudis (2007) presented a more flexible adaptive memory-based algorithm for real-life transportation which was based on the Bone Route method framework for the classical vehicle routing problem. The method was tested on two case studies from the dairy and construction sectors, which were formulated as HFFVRP. Tavakkoli-Moghaddam, Safaei, Kah, and Rabbani (2007) addressed HFFVRP with the split delivery. The authors developed a hybrid-simulated annealing method that was tested on several new instances with the sizes between 6 and 100 customers. Tarantilis, Zachariadis, and Kiranoudis (2008) presented a Guided Tabu Search for solving the HFFVRP. This method was based on Tabu search with a continuous guiding mechanism that directed the search by modifying the problem objective function. Paraskevopoulos, Repoussis, Tarantilis, Ioannou, and Prastacos (2008) proposed a reactive variable neighborhood Tabu search algorithm for solving the HFFVRP with time windows. Yazgi (2010)
applied a visual interactive approach based on a greedy randomized adaptive memory programming search algorithm for solving the HFFVRP and the HFFVRP with backhauls. This algorithm was implemented within a visual interactive decision system, namely ADVISER. Li, Tian, and Anjea (2010) proposed a multi-start adaptive memory programming (MAMP) and path re-linking algorithm to solve the HFFVRP. At each iteration, MAMP constructed several initial solutions and then improved them by a modified Tabu search. Path re-linking was also combined with MAMP in order to enhance its performance. Brandão (2011) has proposed Tabu search algorithm to solve the HFFVRP. Subramanian, Penna, Uchoa, and Ochi (2012) presented a hybrid algorithm, that is composed by an iterated local search (ILS)-based heuristic and a set partitioning (SP) formulation. The SP model is built using routes generated by ILS and it is solved by means of a mixed integer programming solver that interactively calls the ILS heuristic during its execution. Recently, Naji-Azimi and Salari (2013) suggested a general method which is based on ILP and heuristic techniques to improve the quality of a given initial solution. They have used routes obtained by Tarantilis et al. (2003, 2004) and Li et al. (2007) as an initial feasible solution, and tried to improve the quality of the solutions.

Due to the complexity of the HFFVRP, only heuristic and metaheuristic algorithms may efficiently solve this problem. Adaptive memory-based algorithms are very successful metaheuristics for solving combinatorial problems such as VRPs. The aim of this paper is to present an efficient adaptive memory-based algorithm equipped with diversification and intensification mechanisms. This metaheuristic algorithm is based on Bone-Route method which was first proposed for solving the classic VRP by Tarantilis and Kiranoudis (2002) and was later modified in Tarantilis and Kiranoudis (2007) to solve HFFVRP. Bone-Route is a population-based method constructing a new solution using components of routes in the adaptive memory. In this work, a reactive version of this method was presented which directly produced a new solution from the component of other solutions. The proposed method employed the generalized route construction algorithm (GEROCA), which was first presented in Tarantilis and Kiranoudis (2007) for generating initial diversified solutions and modified Tabu search metaheuristic as an improvement procedure. For some valuable applications, see Chen-Fu, Muh-Cherng, and Keng-Han (2013), Ogita and Mori (2012), and Naama, Bouzeboudja, and Allali (2013).

The main contribution of this paper is providing a reactive adaptive memory metaheuristic that makes use of the Tabu Search as an improvement procedure to solve a special class of the vehicle routing problem namely HFFVRP.

The rest of paper is organized as follows: In Section 2, the HFFVRP definition and mathematical formulation are described. In Section 3, first, Tabu search algorithm is presented as an important part of the metaheuristic; then, the purposed algorithm is proposed. Section 4 presents the details of benchmark problems and computational results.

2. Problem statement

2.1. Problem definition

The HFFVRP is defined as follows: let $G = (V, E)$ be an undirected connected graph with a vertex set $V = \{v_0, v_1, \ldots, v_n\}$ and an edge set $E = \{(v_i, v_j) : 0 \leq i, j \leq n\}$. Vertex $v_0$ represents the depot and the other vertex $v_i \in V \setminus \{v_0\}$ is a customer with a non-negative demand $q_i$. With each arc, $(v_i, v_j) \in E$ is associated with the distance $d_{ij}$.
The available fleet consists of \( K \) different types of vehicles located at the depot. A capacity \( Q_k \), a fixed cost \( f_k \), and a variable cost \( a_k \) are associated to each type of vehicle \( k \) and the number of available vehicles of each type is fixed and equal to \( n_k \). \( a_k \) is cost per unit of distance corresponding to each vehicle type \( k \); therefore, \( c_{ij}^k = d_{ij} / a_k \) represents the cost of the travel from customer \( v_i \) to \( v_j \) with a vehicle type \( k \). The aim is to design a set of vehicle routes such that:

- Total transportation cost including the fixed and variable cost is minimized.
- Only available vehicles is used, meaning that no additional vehicles can be considered.
- Each vehicle leaves and returns to the depot.
- The demand of each customer is satisfied by exactly one vehicle in only one visit.
- The total load of each vehicle cannot exceed its capacity.

### 2.2. Problem formulation

The following mathematical formulation is presented for HFFVRP using variables \( x_{ij}^k \) and \( y_{ij}^k \) where \( x_{ij}^k \) takes value 1 if a vehicle of type \( k \) travels directly from customer \( v_i \) to customer; otherwise, 0 denotes the route. The flow variables \( y_{ij}^k \) specify the quantity of goods that a vehicle \( k \) is carrying while leaving customer \( v_i \) to the service customer \( v_j \).

\[
\text{Minimize} \quad \sum_{k=1}^{K} f_k \sum_{j=1}^{n} x_{0j}^k + \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij}^k x_{ij}^k
\]  

subject to

\[
\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^k = 1, \quad \forall j = 1, 2, \ldots, n
\]  

\[
\sum_{i=0}^{n} x_{ij}^k - \sum_{i=0}^{n} x_{ji}^k = 0, \quad \forall i = 1, 2, \ldots, n
\]

\[
\sum_{j=1}^{n} x_{0j}^k \leq n_k, \quad \forall k = 1, 2, \ldots, K
\]

\[
\sum_{k=1}^{K} \sum_{i=0}^{n} y_{ij}^k - \sum_{k=1}^{K} \sum_{i=0}^{n} y_{ji}^k = q_j, \quad \forall j = 1, 2, \ldots, n
\]

\[
q_j x_{ij}^k \leq y_{ij}^k \leq (Q_k - q_i) x_{ij}^k, \quad \forall i, j = 0, 1, \ldots, n, i \neq j
\]

\[
y_{ij}^k \geq 0, \quad \forall i, j = 0, 1, \ldots, n
\]

\[
y_{ij}^k \geq 0, \quad \forall k = 1, 2, \ldots, K
\]

\[
x_{ij}^k \in \{0, 1\}, \quad \forall i, j = 0, 1, \ldots, n
\]

\[
x_{ij}^k \in \{0, 1\}, \quad \forall k = 1, 2, \ldots, K
\]
The objective function (1) gives the sum of total fixed cost of the used vehicles plus the total variable routing cost. Constraint (2) makes sure that each customer is visited exactly once; constraint (3) states that a vehicle of the same type as the one visiting a customer will also depart from it. The maximum number of vehicles available for each vehicle type is guaranteed by constraint (4). Equality (5) insures that all customers’ demand is fully satisfied. Constraint (6) states that the vehicle capacity is never exceeded. Restriction (7) forces that the flow is non-negative. Finally, constraint (8) describes that each arc in the network has the value 1 if it is used; otherwise, it is 0.

3. Solution method

3.1. Tabu search

TS is a memory-based local search optimization algorithm firstly proposed by Glover (1986). As a local search procedure, TS moves from current solution to the next one by exploring the current solution’s neighborhoods. In contrast to the steepest descent approach which stops in a non-improving neighborhood, TS moves from the current solution to the best solution in its neighborhood. To prevent from cycling, it uses a short-term memory known as the Tabu list, which keeps track of the recently visited solutions and avoids moving toward them for a specific number of iterations (Tabu Tenure). Therefore, the neighborhood of the current solution is limited to the solutions which are not declared Tabu or solutions that satisfy the aspiration condition. ‘Aspiration conditions’ are a set of rules, which repeal the Tabu conditions in order to ensure that certain favorable local moves are accepted. In other words, the Tabu restriction may be overridden if the current move leads to a solution that is better than the ones found in the past. A successful application of Tabu search needs a powerful technique for search intensification and diversification. Intensification is comprehensive exploration of some areas of the solution space, typically in the neighborhood of a good solution. Diversification leads the search to the promising regions of the solution space that has not been explored yet.

In order to put the method of solving HFFVRP into practice, the following concepts were defined and used.

Move: A move characterizes the process of generating a feasible solution that is based on the current solution. Practically, the procedures such as the two-Opt move (Croes, 1958), the 1−1 Exchange move (Waters, 1987), and the 1−0 Exchange moves (Waters, 1987) were used on both single route and multiple route processes.

Neighborhood: A solution neighborhood is the collection set of all possible transformations out of a current configuration. In practice, it is difficult to gain the whole neighborhood of a solution and only a subset is usually selected. Since TS accepts non-improving solutions, it is very determinative to investigate how this subset must be selected. In this work, fixed size neighborhood of high-quality solutions was generated by applying the nearest neighborhood algorithm on each solution, which was generated via a single local move. In particular, at any Tabu search iteration, the type of move is chosen randomly. In addition, a predefined probability level is associated to each type of move. Then, it is decided whether to perform the move operation within a single route or between different routes in an unsystematic manner for a more time. This time, for both operations, the probability level is 50%. This neighborhood structure plays a key role in the progress of this metaheuristic method and helps the search to reconnoiter the high-quality solution in the whole solution space.
Tabu list: To avoid the search from the previously visited repeal paths, a Tabu list is established. Any operations that repeal the effect of recent moves were avoided. Due to the storage inefficiency and computational effort, the proposed TS algorithm does not record complete solutions in the Tabu list but the attributes of operations such as move type, the involve nodes, and routes are held. These moves are declared Tabu for a specific number of iterations (Tabu list size) unless the aspiration criterion overrides the Tabu status. In the present approach, if a move from Tabu list resulted in a better cost for all previous iterations, then it would be allowed. The selected aspiration criterion would avoid missing good solutions; therefore, it would not lock the algorithm in the neighborhood of some local minimums.

Long-term memory: The Tabu search step of this method operated on an influence-based long-term memory achieved both diversification and intensification of the search (Tarantilis & Kiranoudis, 2007).

3.2. ReAMTS Algorithm
Here, the reactive adaptive memory-based algorithm (ReAMTS) which used modified Tabu search metaheuristic as an improvement procedure is presented. This approach is based on the Bone Route method proposed by Tarantilis and Kiranoudis (2002). The fact behind the Bone Route is to extract bones (sequences of nodes) with the predefined size and frequency from the adaptive memory (AM). Then, the bones are used to construct new solutions. Common customers cannot be in extracted bones. Therefore, the ReAMTS extracted the bone from the route of the highest quality solutions. The size and frequency specified by the algorithm-designer restrict the number of nodes in a bone (bone-size) and the minimum number of stored routes in the AM which include a bone (bone-freq-min).

In this approach, three main modifications were made with respect to the Bone Route method:

- In order to extract the bones from AM, they must satisfy an additional criterion that specifies the maximum number of stored routes in the AM including a bone (bone-freq-max).
- Values of bone-size and bone-freq-max systematically change during the run time.
- At each iteration of the ReAMTS, except the bones detected from the adaptive memory (AM), some promising bone-like sequences of customers are extracted. The first customer of each sequence is randomly selected and others are determined by the nearest neighborhood heuristic (Step 4 in Figure 1). Then, bone-like sequences and bones are used to construct new solution (Step 6 in Figure 1). This technique drives the search towards promising regions of the solution space.

The values of the bone-freq-max and bone-size express the degree of similarity among the new constructed solution and the previously stored solutions in the AM. Therefore, the new solution is more similar to other solutions in AM whenever their value is high.

Changing the value of these parameters may serve as a strong diversification technique and work as follows: if the ReAMTS does not update the best solution for a pre-specified number of consecutive iterations, \( I_{\text{generalfail}} \), then, the search must be driven towards a part of solution space that has not been explored yet (diversification policy). Therefore, the similarity among the solutions in AM was decreased with the decrease in the value of bone-size and bone-freq-max in a number of consecutive
Reactive Adaptive Memory Based Tabu Search Algorithm

Parameters’ definition

| Parameter       | Definition                                                                 |
|-----------------|-----------------------------------------------------------------------------|
| ISN             | The number of initial solutions                                            |
| Bone-size       | The number of nodes that must compose a bone (bone-length)                 |
| Bone-freq-min   | The minimum number of stored solutions in the AM that include a bone in their routes |
| Bone-freq-max   | The maximum number of stored solutions in the AM that include a bone in their routes |
| $I_{term}$      | The number of iterations in which the search has failed to reach a new and better solution and has terminated |
| $I_{gen/diver}$ | The number of consecutive iterations with diversification policy           |
| $I_{gen/intef}$ | The number of consecutive iterations with intensification policy           |
| $I_{gen/fail}$  | The maximum number of consecutive iterations allowed when the best solution has not been updated |

Variable definition

- **generalDiver**: Is equal to one if we are in Diversification step; otherwise, zero.
- **generalIntef**: Is equal to one if we are in intensification step; otherwise, zero.

Initialization

0: sort vehicles by descending capacities and customers by descending demands

Set: $\text{generalDiver}=0$, $\text{generalIntef}=0$, $gf\text{fail}_\text{itr}=0$, $gf\text{intef}_\text{itr}=0$, $g\text{diver}_\text{itr}=0$, $itr=0$, $fail$ _itr=0.

Set $\text{AM} = \hat{\Phi}$.

1: Generate ISN different diversified initial solutions by employing the (GEROCA) Algorithm.

2: Improve the quality of solutions produced in Step 1 using a Tabu search meta-heuristic and determine the best solution $s^*_\text{in}$ ($s^*_\text{b}=s^*_\text{in}$).

3: Construct the Adaptive Memory (AM). Sort the routes by increasing costs of their relative solutions.

Repeat

4: Extract the promising bone-like sequences of customers according to some selection criteria.

5: Extract the selected bones from the AM according to the values of the bone-size, bone-freq-min and bone-freq-max parameters.

6: Employ the GEROCA constructive heuristic to generate a new solution using the extracted bones and bone-like sequences.

7: Improve the quality of the newly constructed solution generated in Step 6 by a Tabu search and obtain $s_{itr}$.

8: if $\Phi (s_{itr})<\Phi (s^*_\text{in})$ then $s^*_\text{in}=s_{itr}$. Else

   if($fail$ _itr<$I_{term}$)

   $fail$ _itr=$fail$ _itr+1.

   Else exit ();

Figure 1. Outline of the ReAMTS metaheuristic.
9: if \( c(s^*_m) < c(s^*_{m-1}) \) then apply Tabu search meta-heuristic on the best solution \( s^*_m \) and update \( s^*_m \) and AM (if necessary).

10: update AM.

11: (update parameters)

If (!generalDiver && !generalIntef) 

If (\( c(s^*_m) \leq c(s^*_{m-1}) \))

\( gfail_itr = 0. \)

else

if (\( gfail_itr \leq I_{generalfail} \))

\( gfail_itr = gfail_itr + 1. \)

else

Decrease the value of bone-size and bone-freq-max.

\( generalDiver = 1. \)

\( gfail_itr = 0. \)

End if.

End if.

else if (generalDiver)

if (\( c(s^*_m) < c(s^*_{m-1}) \))

if (\( gDiver_itr \leq I_{generalDiver} \))

\( gDiver_itr = gDiver_itr + 1. \)

Else

Increase the value of bone-size and bone-freq-max.

\( gDiver_itr = 0. \)

\( generalDiver = 0. \)

\( generalIntef = 1. \)

end if.

end if.

end if.

Else if (generalIntef)

If (\( c(s^*_m) < c(s^*_{m-1}) \))

if (\( Intef_itr \leq I_{generalIntef} \))

\( glIntef_itr = glIntef_itr + 1. \)

Else

Update the bone-size and bone-freq-max parameters.

\( gintef_itr = 0. \)

\( generalIntef = 0. \)

End if.

End if.

End if.

End if.

Itr = Itr + 1.

Until CPU_time < Max_time;

Figure 1. (Continued)
iterations, \(I_{\text{general_diver}}\). After the diversification policy, the search process was intensified by increasing the values of bone-size and bone-freq-max for a number of consecutive iterations, \(I_{\text{general_intef}}\). After the intensification, the parameters were restored to their initial values. This type of reactive behavior of the adaptive memory was presented in this work for the first time, and has provided a diversified solution to explore the solution space more precisely (Step 11 in Figure 1).

The ReAMTS implementation started from the initial diversified solutions obtained by the GEROMCA (Tarantilis & Kiranoudis, 2007) and improved by using Tabu search.

At each iteration, a new solution was generated by combining the extracted bones from AM and randomly selected promising bone-like sequences of customers. The solution then improved by applying the Tabu search. If AM was not full, the new improved solution would be added, Otherwise, AM was updated by inserting the routes of the newly improved solution and removing the routes belonging to the worst solution if the newly improved solution was better than the lowest quality solution in AM (Step 8 in Figure 1).

At each iteration that the metaheuristic found a better solution, its neighborhood would be explored comprehensively again using the Tabu search algorithm. If TS algorithm successfully improved the solution, adaptive memory would be updated by inserting the routes of the newly improved solution and removing the routes belonging to the previous best solution (Step 9 in Figure 1). Figure 1 shows the main steps of the ReAMTS metaheuristic.

4. Computational results

The proposed metaheuristic presented in Section 3 was programmed in Visual C++ 5.02 and executed on a Pentium IV 2.6 GHz machine with 1.99 Mb of RAM running Windows XP. The performance of the ReAMTS was tested on three sets of benchmark problems for the HFFVRP. In this section, first, the benchmark problems are introduced and then the parameter setting of the ReAMTS algorithm and the details of the obtained computational results are discussed.

4.1. Benchmark problems

The computational work was carried out on three sets of benchmark problems. The first set (Table 1) was taken from Taillard (1999) and contained eight problems, numbered from 13 to 20. These problems were built of 50–100 nodes including the depot, all of which were randomly located over a square with no service time. They had fixed fleet with capacity restrictions without route length. We also tested ReAMTS in the five large size concentric instances from Li et al. (2007), identified as H1–H5, the number of customers in set 2 is between 200 and 360 customers. In these cases, there are only two different values for the demand and all the customers are located at symmetric positions around the depot in concentric circles. Euclidean distances were used in all problems. Table 2 describes the characteristics of these instances.

In order to compare the performance of ReAMTS with an exact method, a new set of sixteen test problems, identified as P1–P16, that had 20–35 customers was generated. The specifications of these problems are reported in Table 3. The data for the depot and the customers (demand and coordinates) were taken from the first set of examples. The specifications were set in such a way that the total capacity of the imposed fleet exceeded the total demand quantities by less than twice the capacity of the smallest
vehicle types. The average capacity ratio of problems in set two was a little greater than that of set one.

4.2. Parameter settings

Metaheuristics must consider several trade-offs; the first being a trade-off between optimality and runtime. This is a trade-off inherent to heuristics as well, but the worst-case ratio for metaheuristics is unknown. Metaheuristics must also consider a trade-off between exploration and exploitation of the search space. Exploration is the capability of an algorithm to discover the majority of the search space while exploitation is the ability to improve a specific solution. An algorithm that explores a large portion of the search space is more likely to find an optimal solution but takes longer to meet. An algorithm that exploits a solution space quickly converges but may be trapped in local optima. Usually, the behavior of a metaheuristic can be changed by adjusting parameters. The solutions produced by the ReAMTS like every metaheuristic algorithm depend on the seed used to generate the sequence of pseudo-random numbers and on different values of the search parameters of the algorithm. It should be mentioned that there was no way for defining the most effective values of the parameters. Therefore, without conducting a difficult sensitivity analysis and statistical tests, general sense

| Problem | $n$ | $k$ | $Q_k$ | $F_k$ | $\alpha_k$ | $n_k$ | Ratio% |
|---------|-----|-----|-------|-------|------------|-------|-------|
| 13      | 50  | 1   | 20    | 20    | 1          | 4     | 95.39 |
|         | 2   | 30  | 35    | 1.1   | 2          |       |       |
|         | 3   | 40  | 50    | 1.2   | 4          |       |       |
|         | 4   | 70  | 120   | 1.7   | 4          |       |       |
|         | 5   | 120 | 225   | 2.5   | 2          |       |       |
|         | 6   | 200 | 400   | 3.2   | 2          |       |       |
| 14      | 50  | 1   | 120   | 100   | 1          | 4     | 88.45 |
|         | 2   | 160 | 1500  | 1.1   | 2          |       |       |
|         | 3   | 300 | 3500  | 1.4   | 1          |       |       |
| 15      | 50  | 1   | 50    | 100   | 1          | 4     | 94.76 |
|         | 2   | 100 | 250   | 1.6   | 3          |       |       |
|         | 3   | 160 | 450   | 2     | 2          |       |       |
| 16      | 50  | 1   | 40    | 100   | 1          | 2     | 94.76 |
|         | 2   | 80  | 200   | 1.6   | 4          |       |       |
|         | 3   | 140 | 400   | 2.1   | 3          |       |       |
| 17      | 75  | 1   | 50    | 25    | 1          | 4     | 95.38 |
|         | 2   | 120 | 80    | 1.2   | 4          |       |       |
|         | 3   | 200 | 150   | 1.5   | 2          |       |       |
|         | 4   | 350 | 320   | 1.8   | 1          |       |       |
| 18      | 75  | 1   | 20    | 10    | 1          | 4     | 95.38 |
|         | 2   | 50  | 35    | 1.3   | 4          |       |       |
|         | 3   | 100 | 100   | 1.9   | 2          |       |       |
|         | 4   | 150 | 180   | 2.4   | 2          |       |       |
|         | 5   | 250 | 400   | 2.9   | 1          |       |       |
|         | 6   | 400 | 800   | 3.2   | 1          |       |       |
| 19      | 100 | 1   | 100   | 500   | 1          | 4     | 76.74 |
|         | 2   | 200 | 1200  | 1.4   | 3          |       |       |
|         | 3   | 300 | 2100  | 1.7   | 3          |       |       |
| 20      | 100 | 1   | 60    | 100   | 1          | 6     | 95.92 |
|         | 2   | 140 | 300   | 1.7   | 4          |       |       |
|         | 3   | 200 | 500   | 2     | 3          |       |       |
parameter-tuning rules, which were easy to understand and could be followed by practitioners, were used. For example:

- The number of nodes that must compose a bone (bone-size) directly depends on the number of customers, n. The initial values of the bone-size and bone-freq-max were determined by the numerical experiments. During the search, bone-size and bone-freq-max changed dynamically in the following way. As explained previously in Section 3.2, if the ReAMTS metaheuristic does not update a best solution for Igeneralfail iterations, then the value of bone-size and bone-freq-max would decrease by 2 units in a Igeneraldiver iterations. After the diversification policy, the value of these parameters increased by 3 units in a Igeneralinter iterations. After the intensification, the parameters were restored to their initial values.

- The number of initial solutions (ISN) and number of solutions stored in adaptive memory (AMsize) determine the trade-off between the quality of the obtained solution and the required computational attempt. The value of ISN is less than AMsize. In Step 1, the goal was to generate a collection of different diversified solutions that may have low quality and need to improve by Tabu search. Therefore, to save time, it was better to spend less time producing different initial solutions and try to improve the existing solutions and construct the Adaptive Memory AM by improved solutions during the search.

| Problem | n  | k  | qk | αk | nk | Ratio% |
|---------|----|----|----|----|----|--------|
| H1      | 200| 1  | 50 | 1  | 8  | 93.02  |
|         | 2  | 100| 1.1| 6  |    |        |
|         | 3  | 200| 1.2| 4  |    |        |
|         | 4  | 500| 1.7| 3  |    |        |
|         | 5  | 1000| 2.5| 1  |    |        |
| H2a     | 240| 1  | 50 | 1  | 10 | 96.00  |
|         | 2  | 100| 1.1| 5  |    |        |
|         | 3  | 200| 1.2| 5  |    |        |
|         | 4  | 500| 17 | 4  |    |        |
|         | 5  | 1000| 2.5| 1  |    |        |
| H3      | 280| 1  | 50 | 1  | 10 | 93.33  |
|         | 2  | 100| 1.1| 5  |    |        |
|         | 3  | 200| 1.2| 5  |    |        |
|         | 4  | 500| 17 | 4  |    |        |
|         | 5  | 1000| 2.5| 2  |    |        |
| H4      | 320| 1  | 50 | 1  | 10 | 94.12  |
|         | 2  | 100| 1.1| 8  |    |        |
|         | 3  | 200| 1.2| 5  |    |        |
|         | 4  | 500| 1.7| 2  |    |        |
|         | 5  | 1000| 2.5| 2  |    |        |
|         | 6  | 1500| 3 |    |    |        |
| H5a     | 360| 1  | 50 | 1  | 10 | 92.31  |
|         | 2  | 100| 1.2| 8  |    |        |
|         | 3  | 200| 1.5| 5  |    |        |
|         | 4  | 500| 1.8| 1  |    |        |
|         | 5  | 1500| 2.5| 2  |    |        |
|         | 6  | 2000| 3 |    |    |        |

*aUsing the values presented in Brandão (2011). See Brandão (2011, p. 146) for more details.
Table 3. Data for problem set 3.

| Problem | $n$ | $k$ | $Q_k$ | $F_k$ | $\alpha_k$ | $N_k$ | Ratio% |
|---------|-----|-----|-------|-------|------------|-------|--------|
| P1      | 20  | 1   | 20    | 20    | 1          | 1     | 93.95  |
|         |     | 2   | 35    | 40    | 1.1        | 2     |        |
|         |     | 3   | 50    | 70    | 1.2        | 2     |        |
|         |     | 4   | 120   | 200   | 2          | 2     |        |
| P2      | 20  | 1   | 40    | 60    | 1          | 1     | 98.33  |
|         |     | 2   | 80    | 100   | 1.6        | 2     |        |
|         |     | 3   | 160   | 250   | 2          | 1     |        |
| P3      | 20  | 1   | 40    | 70    | 1          | 1     | 98.33  |
|         |     | 2   | 80    | 120   | 1.6        | 1     |        |
|         |     | 3   | 120   | 200   | 2.1        | 2     |        |
| P4      | 20  | 1   | 40    | 50    | 1          | 2     | 88.50  |
|         |     | 2   | 80    | 80    | 1.6        | 1     |        |
|         |     | 3   | 120   | 200   | 2.1        | 2     |        |
| P5      | 25  | 1   | 25    | 35    | 1          | 2     | 96.27  |
|         |     | 2   | 35    | 50    | 1.1        | 2     |        |
|         |     | 3   | 50    | 75    | 1.2        | 3     |        |
|         |     | 4   | 120   | 150   | 1.7        | 2     |        |
| P6      | 25  | 1   | 40    | 60    | 1          | 3     | 96.36  |
|         |     | 2   | 80    | 75    | 1.6        | 2     |        |
|         |     | 3   | 160   | 200   | 2          | 1     |        |
| P7      | 25  | 1   | 40    | 60    | 1          | 4     | 88.33  |
|         |     | 2   | 80    | 075   | 1.6        | 2     |        |
|         |     | 3   | 160   | 200   | 2          | 1     |        |
| P8      | 25  | 1   | 30    | 45    | 1          | 2     | 96.36  |
|         |     | 2   | 80    | 120   | 1.6        | 3     |        |
|         |     | 3   | 140   | 210   | 2.1        | 1     |        |
| P9      | 30  | 1   | 25    | 40    | 1          | 3     | 71.51  |
|         |     | 2   | 35    | 60    | 1.1        | 2     |        |
|         |     | 3   | 50    | 80    | 1.2        | 4     |        |
|         |     | 4   | 120   | 200   | 1.7        | 4     |        |
| P10     | 30  | 1   | 120   | 250   | 1          | 2     | 98.26  |
|         |     | 2   | 160   | 400   | 1.6        | 1     |        |
|         |     | 3   | 200   | 600   | 2          | 1     |        |
| P11     | 30  | 1   | 60    | 100   | 1          | 3     | 93.26  |
|         |     | 2   | 100   | 250   | 1.6        | 2     |        |
|         |     | 3   | 140   | 350   | 2          | 1     |        |
| P12     | 30  | 1   | 30    | 45    | 1.5        | 3     | 91.51  |
|         |     | 2   | 80    | 80    | 2          | 2     |        |
|         |     | 3   | 140   | 250   | 2.5        | 2     |        |
| P13     | 35  | 1   | 120   | 300   | 1          | 2     | 97.08  |
|         |     | 2   | 160   | 450   | 1.1        | 3     |        |
| P14     | 35  | 1   | 25    | 40    | 1          | 3     | 84.73  |
|         |     | 2   | 35    | 60    | 1.1        | 2     |        |
|         |     | 3   | 50    | 80    | 1.2        | 4     |        |
|         |     | 4   | 120   | 250   | 1.7        | 4     |        |
| P15     | 35  | 1   | 50    | 80    | 1          | 3     | 94.1   |
|         |     | 2   | 100   | 150   | 1.6        | 3     |        |
|         |     | 3   | 160   | 250   | 2          | 1     |        |
| P16     | 35  | 1   | 40    | 75    | 1          | 2     | 95.67  |
|         |     | 2   | 80    | 90    | 1.6        | 3     |        |
|         |     | 3   | 140   | 250   | 2.1        | 2     |        |
The main drawback of meta-heuristics is that they have no defined stopping criterion; the longer the computing time, the higher the probability of finding the global optima, i.e. the probability of finding a better final solution increases with the run time. In addition, the trade-off between optimality and runtime should be also considered. Hence, Max-CPU-time and lterm were determined proportional to the size of the problem.

The size of Tabu list plays an important role in guiding the search. If the TLsize is too small, the search process does not move away from a local optimum. Also, this prevents the Tabu search from exploring a wide range of solution space; therefore, the TLsize should be as small as possible but long enough to allow the heuristic to move away from local optima.

Considering all the above points, the values of all parameters were determined on the first instance of problem set one, i.e. 13 and H1, by the numerical experiments. Then, to determine the value for parameters on the other instances, several alternative values were tested for each parameter while all others were fixed and the ones were selected that gave the best computational results concerning both the quality of the solution and the computational time needed to achieve this solution. It could be experimentally understood that the most influential parameters in ReAMTS that directly affect the final solution quality are the first six parameters in that indicated in Table 4. Therefore, the values of the last five parameters was kept fixed for all problems in each set. After the selection of the final parameters, 10 different runs with the selected parameters were performed for each instance. The results confirmed that the parameter setting worked well. The most influential parameters of the ReAMTS and their values are listed in Table 4.

4.3. Results on benchmark instances

The results obtained from this algorithm on the first set of benchmark instances were compared with those given by the algorithms of Taillard (1999), Tarantilis et al. (2004), Li et al. (2007), Brandão (2011), Subramanian et al. (2012) and Naji-Azimi and Salari (2013) in Table 5. This table gives the following information: the number of customers; the cost of the solution produced by different algorithms and corresponding computation time in seconds (CPU); the obtained results using the specified parameter setting described in Table 4; the percentage deviation of ReAMTS algorithm’s solution from the best-known solution for each Problem; the best known costs from the literature; the average cost of each method and average deviation (Avg. deviation).

As can be seen from Table 5, the proposed algorithm produced the best-known solution for five out of eight problem instances in a reasonable time, as published in the literature.

The results indicated that ReAMTS was a competitive approach compared with the traditional and new metaheuristics. ReAMTS made improvement to the best-known solutions of instances 20 and 16, as much as 11.824 and 1.004%, respectively. In other instances including 13, 14, and 15, the proposed algorithm found the best-known solutions, i.e. the gap was zero. Overall, the ReAMTS in average improved the best solution cost as much as 1.49% for these instances. Moreover, no algorithm was able to obtain the solution quality achieved by the ReAMTS for Problems 16 and 20. Hereunto. Even ILP-based method presented in Naji-Azimi and Salari (2013) was not able to improve the solution of problem 16 and 20 up to our results (see Naji-Azimi & Salari, 2013, pp. 4321–4323).
Table 4. Parameter setting for metaheuristic.

| Parameter         | Description                                                                 | Problem number | Value  |
|-------------------|-----------------------------------------------------------------------------|----------------|--------|
| TLsize            | Size of the Tabu list                                                       | 13–16          | 20     |
|                   |                                                                             | P1–P16         | 30     |
|                   |                                                                             | 17–20          |        |
|                   |                                                                             | H1 & H2        | 50     |
|                   |                                                                             | H3 & H4        | 70     |
|                   |                                                                             | H5             | 100    |
|                   |                                                                             | 13–20          | 10     |
| AMsize            | The number of solutions stored in AM                                         | P1–P16         | 2      |
|                   |                                                                             | H1–H3          | 10     |
|                   |                                                                             | H4 & H5        | 5      |
| ISN               | The number of initial solutions                                              | P1–P16         | 2      |
|                   |                                                                             | H1–H3          | 5      |
|                   |                                                                             | H4–H5          | 2      |
| Bone_size         | The number of nodes that must compose a bone (bone-length)                   | 13–16          | 3      |
|                   |                                                                             | 17 & 18        | 4      |
|                   |                                                                             | 19 & 20        | 5      |
|                   |                                                                             | P1–P16         | 2      |
|                   |                                                                             | H1–H3          | 5      |
|                   |                                                                             | H4 & H5        | 8      |
| Max_CPU-time      | The maximum allowed running time                                             | 13–16          | 100    |
|                   |                                                                             | 17–20          | 300    |
|                   |                                                                             | P1–P16         | 50     |
|                   |                                                                             | H1–H3          | 10,000 |
|                   |                                                                             | H4 & H5        | 15,000 |
| Iterm             | The number of iterations at which the search of the metaheuristic has failed to reach a new and best solution and has been terminated | 13–16         | 150    |
|                   |                                                                             | 17–20          | 200    |
|                   |                                                                             | H1–H3          | 300    |
|                   |                                                                             | H4 & H5        | 500    |
| Bonefreq_min      | The minimum number of stored solutions in the AM that must include a bone in their routes | All problems | 2      |
| Bonefreq_max      | The maximum number of stored solutions in the AM that must include a bone in their routes | All problems | AMsize |
| Igeneral fail     | The maximum number of consecutive iterations allowed when the best solution has not been updated | 13–20          | 10     |
|                   |                                                                             | P1–P16         | 100    |
|                   |                                                                             | H1–H5          |        |
| IgeneralDive      | The number of consecutive iterations with diversification policy             | 13–20          | 5      |
|                   |                                                                             | P1–P16         |        |
|                   |                                                                             | H1–H5          | 30     |
| IgeneralIntef     | The number of consecutive iterations with intensification policy             | 13–20          | 5      |
|                   |                                                                             | P1–P16         |        |
|                   |                                                                             | H1–H5          | 30     |
Table 5. Comparison results for problem set 1.

| Problem | n   | Taillard (1999) | Li et al. (2007) | Brandão (2011) | Subramanian et al. (2012) | Naji-Azimi et al. (2013) | Our results |
|---------|-----|-----------------|------------------|-----------------|---------------------------|---------------------------|-------------|
|         |     | CGA (sec)       | BATA (sec)       | HRTR (sec)      | TSA (sec)                 | ILS-RVND-SP (sec)         | ILP-based   |
| 13      | 50  | 1518.05         | 743              | 1519.96         | 358                       | 1517.84                   | 3.3         |
| 14      | 50  | 615.64          | 573              | 611.39          | 387                       | 607.53                    | 1.09        |
| 15      | 50  | 1016.36         | 335              | 1015.29         | 166                       | 1015.29                   | 2.13        |
| 16      | 50  | 1154.05         | 350              | 1145.52         | 341                       | 1144.94                   | 1.14        |
| 17      | 75  | 1071.79         | 2245             | 1071.01         | 363                       | 1061.96                   | 4.22        |
| 18      | 75  | 1870.16         | 2876             | 1846.35         | 971                       | 1823.38                   | 4.06        |
| 19      | 100 | 1117.51         | 5833             | 1123.83         | 428                       | 1120.34                   | 9.12        |
| 20      | 100 | 1559.77         | 3402             | 1556.35         | 1156                      | 1534.17                   | 8.89        |
| Avg     |     | 1240.42         | 2044.63          | 1236.21         | 607.125                   | 1228.18                   | 8.3         |
| Avg. deviation (%) |     | -1.03           | -.68             | -.03            | .03                       | -.03                      | 1.49        |

Deviation = \( \frac{\text{Best known solution} \times \text{ReAMTS solution}}{\text{Best known solution}} \times 100 \).
More precisely, it can be observed that, in comparison with Taillard (1999), Li et al. (2007) and Brandão (2011), the ReAMTS produced better solutions for all the problems except two. In comparison with Tarantilis et al. (2004), the solutions given for the five problems by the ReAMTS were better. In comparison with Naji-Azimi and salari (2013), it can be observed that the solution given by the ILP-based heuristic in one problem is little better and in problems 16, 17, and 20, results of an ILP-based heuristic on three different initial solutions are not as good as ReAMTS. These results have proven that the metaheuristic is appropriate and the algorithm is able to find high-quality solutions in a reasonable computing time.

However, as noted in Tarantilis et al. (2008), the direct comparison of the required computational time cannot be conducted as they closely depend on various factors such as the processing power of the computers, the programming languages, the coding abilities of the programmers, the compilers, and the running processes of the computers. A simple criterion to measure the efficiency and the quality of an algorithm is to compute the relation percentage deviation of its solution from the best solution reported in the literature on specific benchmark instances. In Table 6, the percentage of relative deviation of each algorithm’s solution is compared with the best-known solution. From this table, it can be concluded that the ReAMTS method had the least deviation from the best solution.

The results obtained from this algorithm on the second set of benchmark were compared with those given by the algorithms of Li et al. (2007), Brandão (2011), Subramanian et al. (2012) in Table 7. We can see that the ReAMTS method was capable to make improvement in one case and get the same results for two others. In comparison with Subramanian et al. (2012) and Li et al. (2007), the ReAMTS produced better solutions for all problems. In comparison with Brandão (2011) Brandao produced slightly better solutions than ReAMTS. Overall, ReAMTS and Brandão (2011) that uses the Tabu Search has the better results. It is shown that computing time of ReAMTS is higher than other algorithms because it should reassign the customers and extracted bones and bone-likes to the limited number of vehicles to construct new feasible solution. Also it is more time-consuming when the ratio is close to one, we believe this is acceptable by considering a trade-off between solution quality and time.

Table 8 presents the results of the ReAMTS method over the new 16 benchmark instances and compares them with the results of solver CPLEX on AIMMS software. AIMMS is an advanced development environment for building optimization based operations’ research applications and advanced planning systems (Optimization Software for Operations Research Applications). Moreover, to see how other metaheuristics work with problem set 2, the Tabu search algorithm was programmed (Brandão, 2011), which was newer than others, in Visual C++ 5.02, and executed it on a Pentium IV 2.6 GHz machine with 1.99 Mb of RAM running Windows XP. The results obtained by TS algorithm on the problem set 3 are presented in Table 8. The information in Table 8 consists of the number of customers; the cost of the solution produced by different algorithms and respective computing time in seconds (CPU) and respective percentage of relative deviation of each algorithm’s solution compared with the best- solution obtained; the ratio of total demand to total capacity.

AIMMS with solver CPLEX obtained an optimal solution only for P1, P4, and P9 and, in other instances; it was automatically terminated before reaching the optimal solution. It is worth nothing that the ratio of H1, H4, and H9 is less than other problems with the same size; Also, in problem 15, it failed to obtain a feasible solution. As can be seen from Table 8, the ReAMTS found an optimal solution for problems H1, H4,
Table 6. Comparison results of percentage deviation for problem set 1.

| Problem | $n$ | Best known solution | Taillard CGA | Tarantlis et al. BATA | Li et al. HRTR | Brandao TSA | Subramanian et al. ILS-RVND-SP | Naji-Azimi et al. ILP-based method | Our algorithm AMTS |
|---------|----|---------------------|-------------|------------------------|--------------|-------------|-----------------------------|---------------------------------|------------------|
| 13      | 50 | 1517.84$^a$         | −.01        | −.14                   | .00          | .00         | .000                        | .000                            | .00              |
| 14      | 50 | 607.53$^b$          | −1.33       | −.63                   | .00          | .00         | .000                        | .000                            | .00              |
| 15      | 50 | 1015.29$^d$         | −.10        | .00                    | .00          | .00         | .000                        | .000                            | .00              |
| 16      | 50 | 1133.44$^b$         | −1.82       | −1.06                  | −1.01        | −1.01       | −1.015                      | −1.015                          | .00              |
| 17      | 75 | 1061.96$^b$         | −.92        | −.85                   | .00          | .00         | .000                        | .000                            | −1.89            |
| 18      | 75 | 1823.38$^a$         | −2.57       | −1.26                  | .00          | −.44        | .000                        | −.768                           | −1.29            |
| 19      | 100| 1117.51$^c$         | .00         | −.56                   | −.25         | −.25        | −.253                       | −.253                           | −.25             |
| 20      | 100| 1352.77$^b$         | −15.30      | −15.05                 | −13.41       | −13.41      | −13.410                     | −14.099                         | .00              |
| Avg     |    | 1203.76$^b$         | −2.76       | −2.44                  | −1.59        | −1.59       | −1.835                      | −2.017                          | −.43             |

$^a$Solution cost taken from Li et al. (2007).
$^b$Found by ReAMTS.
$^c$Solution cost taken from Taillard (1999).
$^d$Solution cost taken from Tarantlis et al. (2004).
| Problem | n   | Li et al. (2007) | Brandão (2011) | Subramanian et al. (2012) | Our results |
|---------|-----|-----------------|----------------|---------------------------|-------------|
|         |     | HRTR Time (sec) | TSA Time (sec) | ILS_RVND_SP Time (sec)   | ReAMTS Time (sec) | Deviation (%) | Best known solution |
| H1      | 200 | 12067.65        | 687.82         | 12050.08                  | 12050.2      | 1254.7        | .005 12050.8        |
| H2      | 240 | 10234.40        | 995.27         | 10208.32                  | 10209.3      | 3283.1        | -.01 10208.32       |
| H3      | 280 | 16231.80        | 1437           | 16223.39                  | 16282.41     | 259.61        | 1.76 16232.39       |
| H4      | 320 | 17576.10        | 2256.35        | 17458.65                  | 17458.65     | 6269.5        | 0 17458.65         |
| H5      | 360 | –               | –              | 23166.56                  | 23168.4      | 12157.6       | -.008 23166.56      |
| Avg     | –   | 14027.49        | 1344.11        | 13985.29                  | 13987.59     | 5219.38       | -.01 13985.29       |
| Deviation (%) | – | -.30 | – | .00 | – | -.83 | – | -.02 | – | – | .00 |
Table 8. Comparison results for problem set 3.

| n  | Best found solution | ReAMTS Cost | Time (sec) | Deviation (%) | TS | Time2 (sec) | Deviation (%) | AIMMS Cost | Time3 (sec) | Deviation2 (%) | Ratio (%) |
|----|---------------------|-------------|------------|--------------|----|------------|--------------|------------|------------|----------------|-----------|
| P1 | 20                  | 616.429     | .625       | .000         | 616.429 | .402       | .000         | 616.429    | 5716.220   | −3.372         | 93.950 |
| P2 | 20                  | 549.728     | .312       | .000         | 549.728 | 1.937      | .000         | 568.264    | 27183.500  | −3.372         | 93.950 |
| P3 | 20                  | 591.424     | 2.906      | −.093        | 591.424 | 1.671      | .000         | 591.424    | 31843.151  | −3.372         | 93.950 |
| P4 | 20                  | 570.797     | 3.281      | .000         | 570.797 | 3.281      | .000         | 570.797    | 48123.980  | −3.372         | 93.950 |
| P5 | 25                  | 750.185     | 2.657      | .000         | 761.730 | 3.761      | −1.539       | 756.114    | 36106.160  | −.790          | 88.330 |
| P6 | 25                  | 626.238     | 1.656      | .000         | 643.875 | 2.656      | −2.816       | 637.972    | 31338.128  | −1.874         | 86.330 |
| P7 | 25                  | 625.786     | 1.334      | .000         | 634.418 | 3.313      | −1.379       | 631.692    | 33612.924  | −.944          | 88.330 |
| P8 | 25                  | 657.361     | 4.062      | .000         | 681.695 | 2.688      | −3.702       | 666.383    | 36802.110  | −1.372         | 96.360 |
| P9 | 30                  | 730.827     | 2.844      | .000         | 744.300 | 6.328      | −1.844       | 730.827    | 42518.890  | .000           | 71.510 |
| P10| 30                  | 536.847     | 13.625     | .000         | 572.556 | 14.672     | −6.652       | 830.601    | 42518.890  | −54.718        | 98.260 |
| P11| 30                  | 617.510     | 4.000      | .000         | 622.145 | 6.266      | −.751        | 825.263    | 15723.280  | −33.644        | 93.260 |
| P12| 30                  | 918.413     | 9.359      | .000         | 926.911 | 7.719      | −.925        | 927.046    | 43558.700  | −.940          | 91.510 |
| P13| 35                  | 880.573     | 4.968      | .000         | 880.573 | 9.021      | .000         | 883.071    | 29509.980  | −284           | 97.080 |
| P14| 35                  | 486.575     | 1.200      | .000         | 490.842 | 13.687     | −.877        | 518.710    | 33427.880  | −6.604         | 84.730 |
| P15| 35                  | 748.104     | 5.297      | .000         | 793.146 | 10.325     | −6.021       | **45500.890** | 11989.380  | −.940          | 94.100 |
| P16| 35                  | 813.000     | 8.063      | −3.558       | 813.000 | 8.093      | .000         | 1375.489   | 11989.380  | −69.187        | 95.670 |
| Avg|                    | 669.987     | 671.830    | −.228        | 680.848 | −1.657     | —            | 742.005    | —          | −11.582       | —         |
and H9 in much less time and a better solution for other instances. The solutions discovered by ReAMTS were better than those reported in TSA (Brandão, 2011). The ReAMTS produced better solutions for all the problems except two. In problem 16, TSA (Brandão, 2011) found the best solution in comparison with AIMMS and ReAMTS. We also tested AIMMS with solver CPLEX on problem 13 in set 1 and observed that it was automatically terminated before reaching the feasible solution.

From this comparison, it can be also concluded that metaheuristics procedures, even in a small problem and especially when the ratio is close to one, had better performance than the exact methods.

5. Conclusions

As is known, the ReAMTS is the first reactive adaptive memory-based metaheuristic which has been successfully applied for the HFFVRP. The results proved that the method was appropriate and capable of finding high-quality solutions in a reasonable computing time. In average, the ReAMTS had good performance on large-scale problems and made improvement to the best-known solutions of problem set one as much as 1.49%. In addition, its performance on 16 new HFFVRPs problem compared with the well-known software AIMMS with solver CPLEX demonstrated the solution quality.

Finally, we have identified a number of ways in which we might be able to improve the computational performance of the algorithm and apply the method to the other class of problem. These are topics of our future investigations.

References

AIMMS 3.9 Paragon Decision Technology B.V., Netherlands. Retrieved from http://www.AIMMS.com

Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Computers & Operations Research, 38, 140–151.

Chen-Fu, C., Muh-Cherng, W., & Keng-Han, L. (2013). Effect of solution representations on Tabu search in scheduling applications. Computers & Operations Research, 40, 2817–2825.

Croes, G. (1958). A method for solving traveling-salesman problems. Operations Research, 6, 791–812.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management Science, 6, 80–91.

Dueck, G., & Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computational Physics, 90, 161–175.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & Operations Research, 13, 533–549.

Hoff, A., Andersson, H., Christiansen, M., Hasle, G., & Lokketangen, L. (2010). Industrial aspects and literature survey: Fleet composition and routing. Computers & Operations Research, 37, 2041–2061.

Li, F., Golden, B., & Wasil, E. (2005). Very large-scale vehicle routing: New test problems, algorithms, and results. Computers & Operations Research, 1165–1179.

Li, F., Golden, B. L., & Wasil, E. A. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers & Operations Research, 34, 2734–2742.

Li, X., Tian, P., & Aneja, Y. P. (2010). An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 46, 1111–1127.

Naama, B., Bouzeboudja, H., & Allali, A. (2013). Solving the economic dispatch problem by using tabu search algorithm. Energy Procedia, 36, 694–701.
Naji-Azimi, Z., & Salari, M. (2013). A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem. *Applied Mathematical Modelling*, 4316–4324.

Ogita, Y., & Mori, H. (2012). Parallel dual Tabu search for capacitor placement in smart grids. *Procedia Computer Science*, 12, 307–313.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14, 425–455.

Subramanian, A., Penna, P. H. V., Uchoa, E., & Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221, 285–295.

Rochat, Y., & Taillard, E. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1, 147–167.

Tarantilis, C. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO – Operations Research*, 33, 1–14.

Tarantilis, C. D., & Kiranoudis, C. T. (2002). BoneRoute: An adaptive memory-based method for effective fleet management. *Annals of Operations Research*, 115, 227–241.

Tarantilis, C. D., & Kiranoudis, C. T. (2007). A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. *European Journal of Operational Research*, 179, 806–822.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2003). A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Journal of the Operational Research Society*, 54, 65–71.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152, 148–158.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152, 148–158.

Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2008). A guided tabu search for the heterogeneous vehicle routing problem. *Journal of the Operational Research Society*, 59, 1659–1673.

Tavakkoli-Moghaddam, R., Safaei, N., Kah, M. M. O., & Rabbani, M. (2007). A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute*, 344, 406–425.

Waters, C. D. J. (1987). A solution procedure for the vehicle-scheduling problem based on iterative route improvement. *Journal of the Operational Research Society*, 38, 833–839.

Yazgi, T. G. (2010). An interactive GRAMPS algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls. *European Journal of Operational Research*, 201, 593–600.
Appendix A. Specifications of problem set 1 and the best solutions are reported in the following tables

| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost  |
|----------------|------------------|------------|-----------------|-------------------|------------|
| 13             | 200              | 192        | 10              | 0 45 29 5 37 20 36 47 21 48 30 0 | 301.396    |
| 120            | 117              | 6          | 0 12 39 9 32 44 3 0 | 153.647          |
| 120            | 118              | 4          | 0 14 11 38 10 0 | 198.738          |
| 70             | 68               | 3          | 0 28 22 33 0    | 115.271          |
| 70             | 68               | 5          | 0 23 41 42 43 1 0 | 142.691          |
| 70             | 67               | 4          | 0 50 18 24 49 0 | 142.761          |
| 70             | 62               | 3          | 0 34 46 8 0    | 56.1807          |
| 40             | 37               | 3          | 0 15 13 27 0   | 73.7154          |
| 40             | 39               | 2          | 0 25 31 0      | 106.546          |
| 40             | 40               | 3          | 0 19 35 7 0   | 59.798           |
| 40             | 33               | 1          | 0 40 0         | 33.9411          |
| 30             | 26               | 1          | 0 2 0          | 32.0325          |
| 30             | 30               | 1          | 0 4 0          | 15.5563          |
| 20             | 18               | 1          | 0 26 0        | 12.1655          |
| 20             | 20               | 1          | 0 17 0        | 16.1245          |
| 20             | 19               | 1          | 0 6 0        | 18.4391          |
| 20             | 19               | 1          | 0 16 0       | 38.833           |
| Sum            | –                | – 50       | –              | – 1517.84        |

| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost |
|----------------|------------------|------------|-----------------|-------------------|------------|
| 14             | 300              | 300        | 12              | 0 17 40 32 9 39 31 10 38 11 14 19 8 0 | 171.154    |
| 160            | 159              | 10         | 0 45 29 5 47 36 37 20 15 13 27 0 | 107.455    |
| 160            | 156              | 7          | 0 6 2 22 28 21 48 30 0 | 91.8188    |
| 120            | 119              | 6          | 0 4 34 46 35 7 26 0 | 46.2541    |
| 120            | 119              | 8          | 0 33 1 43 42 41 23 49 16 0 | 91.6479    |
| 120            | 120              | 7          | 0 3 44 24 18 50 25 12 0 | 99.1988    |
| Sum            | –                | – 50       | –              | – 607.529     |

| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost |
|----------------|------------------|------------|-----------------|-------------------|------------|
| 15             | 160              | 156        | 10              | 0 32 2 29 21 34 30 9 50 16 11 0 | 190.545    |
| 160            | 145              | 10         | 0 27 8 31 28 3 36 35 20 22 1 0 | 216.601    |
| 100            | 99               | 4          | 0 14 25 13 41 0 | 124.129     |
| 100            | 95               | 3          | 0 12 47 18 0   | 59.0229     |
| 100            | 99               | 8          | 0 49 10 39 33 45 15 44 17 0 | 158.04     |
| 50             | 41               | 3          | 0 5 38 46 0   | 37.0507     |
| 50             | 47               | 4          | 0 24 3 7 26 0 | 88.8114     |
| 50             | 47               | 5          | 0 4 19 40 42 37 0 | 93.4816    |
| 50             | 48               | 3          | 0 48 23 6 0   | 47.613      |
| Sum            | –                | – 50       | –              | – 1015.29    |
| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost |
|---------------|-----------------|------------|-----------------|------------------|------------|
| 17            | 140             | 138        | 5               | 0 47 18 13 25 14 0 | 146.129   |
|               | 140             | 140        | 8               | 0 32 2 29 21 34 30 49 5 0 | 180.327   |
|               | 140             | 137        | 10              | 0 1 22 20 35 36 3 28 31 26 8 0 | 244.29    |
| 80            | 71              | 7          | 6               | 0 10 39 33 45 15 37 0 | 146.045   |
| 80            | 75              | 7          | 6               | 0 11 16 50 9 38 46 0 | 90.3567   |
| 80            | 77              | 7          | 7               | 0 4 41 40 19 42 44 17 0 | 156.828   |
| 80            | 63              | 4          | 4               | 0 27 48 23 6 0 | 77.4463   |
| 40            | 29              | 1          | 1               | 0 12 0 | 16.1245   |
| 40            | 40              | 3          | 3               | 0 24 43 7 0 | 75.8905   |
| Sum           | –               | –          | 50              | – | 1133.44   |
| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes                        | Total cost |
|----------------|------------------|------------|-----------------|------------------------------------------|------------|
| 18             | 400              | 390        | 18              | 0 26 12 40 3 44 32 9 39 72 58 10 38 65 66 11 53 7 67 0 | 392.648    |
| 250            | 250              | 250        | 16              | 0 46 52 27 45 29 5 37 20 70 60 71 69 36 47 48 68 0 | 343.669    |
| 150            | 149              | 149        | 7               | 0 30 74 21 61 28 2 6 0                    | 186.196    |
| 150            | 147              | 147        | 8               | 0 62 22 64 42 41 43 1 33 0                 | 226.354    |
| 100            | 98               | 98         | 6               | 0 51 16 49 24 18 50 0                     | 160.958    |
| 100            | 96               | 96         | 5               | 0 8 19 59 14 35 0                         | 151.759    |
| 50             | 49               | 49         | 2               | 0 34 4 0                                  | 31.3848    |
| 50             | 49               | 49         | 4               | 0 63 23 56 73 0                           | 102.766    |
| 50             | 50               | 50         | 4               | 0 15 57 13 54 0                           | 97.2639    |
| 50             | 46               | 46         | 3               | 0 31 55 25 0                              | 131.832    |
| 20             | 20               | 20         | 1               | 0 17 0                                    | 16.1245    |
| 20             | 20               | 20         | 1               | 0 75 0                                    | 6          |
| Sum            | –                | –          | 75              | –                                         | 1846.96    |
| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost |
|----------------|------------------|------------|----------------|-------------------|------------|
| 19             | 300              | 296        | 16             | 0 13 97 92 37 98 100 91 44 86 16 61 85 93 59 95 94 0 | 146.335    |
| 300            | 278              | 17         | 0 28 26 4 39 67 23 56 75 41 22 74 72 73 21 40 58 53 0 | 198.874    |
| 200            | 199              | 14         | 0 27 69 1 70 30 20 66 32 90 63 10 62 88 31 0 | 159.508    |
| 200            | 193              | 12         | 0 89 18 82 48 47 36 49 64 11 19 7 52 0 | 165.9684   |
| 200            | 199              | 14         | 0 76 77 3 79 78 34 35 65 71 9 51 81 33 50 0 | 166.31     |
| 100            | 96               | 8          | 0 87 42 14 38 43 15 57 2 0 | 102.498    |
| 100            | 98               | 11         | 0 60 83 8 46 45 17 84 5 99 96 6 0 | 92.0161    |
| 100            | 99               | 8          | 0 54 55 25 24 29 68 80 12 0 | 88.834     |
| Sum            | –                | –          | 100            | –                 | 1120.34    |
| Problem number | Vehicle capacity | Total load | Total customers | Sequence of nodes | Total cost |
|----------------|------------------|------------|----------------|-------------------|------------|
| 20             | 200              | 198        | 10             | 0 87 100 44 38 86 16 61 85 98 6 0 | 201.46     |
| 200            | 179              | 9          | 0 7 48 47 49 63 62 88 31 27 0 | 216.922     |
| 140            | 131              | 6          | 0 13 95 59 93 96 94 0 | 75.3456     |
| 140            | 131              | 8          | 0 77 33 81 9 65 66 20 1 0 | 187.307     |
| 140            | 136              | 8          | 0 75 56 23 67 39 25 55 4 0 | 183.041     |
| 140            | 140              | 9          | 0 50 3 79 78 29 68 80 54 12 0 | 149.167     |
| 100            | 58               | 5          | 0 89 5 84 46 8 0 | 80.1243     |
| 60             | 57               | 4          | 0 28 40 58 53 0 | 32.0604     |
| 60             | 58               | 5          | 0 90 32 30 70 69 0 | 73.1823     |
| 60             | 56               | 4          | 0 41 22 74 72 0 | 61.9011     |
| 60             | 60               | 6          | 0 42 14 43 15 73 21 0 | 92.254     |
| Sum            | –                | –          | 100            | –                | 1352.77    |