International Journal of Computer Science and Applications,

Vol. 10, No. 2, pp. 72– 91, 2013

MEMETIC ALGORITHM FOR A MULTI-OBJECTIVE VEHICLE ROUTING PROBLEM WITH MULTIPLE TRIPS

RAJAA AYADI, YOUSSEF BENADADA

Mohammed V University Souissi, ENSIAS, RO_L
Rabat, Morocco

ayadirajaa@gmail.com, yss.benadada@gmail.com

This paper considers a variant of the vehicle routing problem where vehicles are allowed to take more than one route during the working day. A restricted fleet size is used in order to serve customers' demand. Therefore, vehicles could exceed the time horizon. The objective is to optimize both maximum overtime and routing cost. A mathematical model has been proposed for the VRPM. A memetic algorithm has been developed to solve it by combining a genetic algorithm based on suitable encoding and genetic operators, with a local search procedure. The algorithm is run over a benchmark of the VRPM. Comparing with seven existing algorithms, our approach shows competitive performance and contributes many new best solutions.

Keywords: VRP, multiple trips, multi-objective, memetic algorithm.

1. Introduction

This paper deals with the vehicle routing problem with multiple trips (VRPM). This problem is inspired by a real life case of the distribution of Liquefied Petrol Gas (LPG). It consists in optimizing the delivery of a LPG product to a set of customers, which must be supplied in one visit by a limited fleet of homogeneous vehicles. The fleet is initially based in a depot. The objective is to find a complete routing schedule for serving all customers with the minimum routing cost, subject to a number of constraints such as vehicles capacity and legal conductors work time.

Unlike traditional cases where vehicles can do just one route during the problem period, it is advisable in the studied case that vehicles perform several routes in the daytime in order to optimize the use of the fleet (Fig.1).

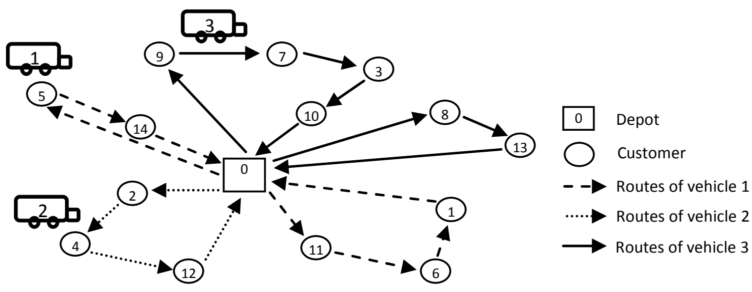

Fig. 1: An example of vehicle routing problem with multiple trips

In the real problem, a large size of the fleet is available to provide good service. However, data sets used by VRPM benchmarks well known in the literature, consider a restricted fleet size. Thus, a feasible solution is not easily obtained, and routes may exceed the time horizon. This constraint violation is called overtime. Benchmarks work with the objective of minimizing the maximum overtime. To be able to compare results with works dealing with similar problem, the same objective is considered; minimizing the maximum overtime in addition to minimizing routing cost, i.e. the total distance. For the considered problem, time and distance are assumed to be proportional and symmetric.

2. Related work

The examined problem is a generalization of the Vehicle Routing Problem (VRP) first introduced by [Dantzig and Ramser, 1959]. The literature concerning Vehicle Routing Problem is abundant. [Eksioglu et al., 2009] present a classification of the VRP literature based on a literature review of about 1500 papers. There are several variants of the VRP; the main ones are presented in [Toth and Vigo, 2001].

Most of distribution problems assume that each vehicle performs one route and returns back to the depot. However, this assumption is not practical in real life; a vehicle carrying a full vehicle load of goods to nearby customers can still have time to return to the depot for replenishment and start another route. Distribution companies seek to minimize vehicle cost by reducing the number of required vehicles and by using vehicles with less cost.

Fleischmann was the first to introduce the multiple trips idea for VRP [Fleischmann, 1990]; since then, several studies have been interested in VRP with multiple trips to solve some real life problems. VRPM was solved, in the literature, following two types of approach: the monolithic approach using heuristics and meta-heuristics, and the decomposition approach that employs two phases; find a solution to VRP first, and then combine single trips to have multiple trips per vehicle.

[Fleischmann, 1990] solve the problem in two phases, he use savings based method and bin packing heuristic. [Taillard, Laporte and Gendreau, 1996] work on the VRPM allowing for overtime, paying some penalty. Their solution is a population based algorithm using tabu search and bin packing approaches. [Golden, Laporte and Taillard, 1996] also present a work on VRPM where they assume that the vehicles have an infinite capacity, and seek to minimize the length of the maximum distance covered by any vehicle at the end of the day. [Brandao and Mercer, 1997] use tabu search to find a solution for a real life firm with several constraints, and then apply the same heuristic to resolve the simple VRPM [Brandao and Mercer, 1998]. [Patch and Salhi, 2004] consider a multi-phase construction heuristic. Then, they use the genetic algorithm for the same problem [Salhi and Petch, 2007]. [Alonso, Alvarez and Beasley, 2008] use a tabu Search Algorithm for a Periodic Vehicle Routing Problem with Multiple Trips and constraint of Site-dependency. [Olivera and Viera, 2007] use an adaptive memory procedure to solve VRPM, and [Şen and Bülbül, 2008] present a survey on multi-trip vehicle routing problem where they report results of most works presented above. Recently, [Mingozzi *et* *al.*, 2012] use an exact algorithm based on two set-partitioning-like formulations of the VRPM and solve to optimality 42 of the 56 benchmark instances tested.

Inspired by real life cases, some works combine multiple trips constraints with other variants of the VRP. Multi-trips VRP with time windows (MVRPTW) have important applications in practice. [Azi et al., 2007] develop an exact column generation approach for a single-vehicle and time window case with small sized instances, and then they extend this work to the multiple vehicle case [Azi et al., 2010]. They also work on a dynamic VRPM using an adaptive large neighborhood search heuristic [Azi et al., 2011]. [Battarra et al., 2009] deals with the strategic fleet sizing in the distribution of multiple products wi th time window constraint. [Hernandez et al., 2010] use a column generation for the same variant with constraint of limit routes' duration. [Macedo et al., 2011] propose a new exact method for solving MVRPTW, and [Prins, 2002] use efficient heuristics for the heterogeneous fleet multi-trip VRP for a real case problem.

Multiple trips may also influence strategic decisions of an enterprise. As a matter of fact, [Lin and Kwok, 2006] solve a problem of location-routing problem with multiple uses of vehicles, using tabu search and simulated annealing meta-heuristics. [Huang and Lee, 2011] attempt to solve both the VRPM and the distribution center location problem at the same time, they use a Simulated Annealing based algorithm. [Derigs et al., 2011] present two approaches based on a meta-heuristic guided neighborhood search for solving a real-world vehicle routing problem that combines transportation tasks and trips performed by tractors. The objective is to minimize the number of required tractors.

This work deals with the VRPM problem as it has been described by [Taillard, Laporte and Gendreau, 1996] and uses benchmark problems that they propose. A mathematical model is proposed for the problem. To solve it, a meta-heuristic based on the genetic algorithm and local search is implemented. A new efficient encoding is proposed for the genetic representation of the multiple trips variant of VRP.

Unlike some decomposition methods where routes are assigned to vehicles at the last step, vehicle allocation to routes is done here during the construction. This approach helps to minimize overtime and find feasible solutions that respect time horizon. Performance results are obtained.

3. The mathematical model

This model uses a three index formulation for decision variable x_{ij}^{r} which indicates whether vehicle of route r goes from node i to node j or not. The notation used for the other variables is presented below:

d_{ij} : Routing cost between i and j

t_{ij} : Time of travelling from i to j

Q : Capacity of vehicles

T : Maximum driving time

αT : Maximum legal overtime

q_i : Quantity demanded by customer i

K : Number of vehicles

N : Number of customers

n : Maximum number of routes that can be assigned to a vehicle during the day

0 : Index of depot

1 to N : Index of customers

It is assumed that each vehicle could serve a maximum of n routes during day. So, the maximum number of possible routes is nK. For this purpose, S_k is defined to be the set of the n routes assigned to vehicle k, we note:

$$S_k = \{k + qK, q = 0 \,..\, n-1\} \tag{1}$$

For example, if n=3 then vehicle k can follow the routes k, $k+K$ and $k+2K$.A route r of S_k is not null and it is really performed by the vehicle k if there is at least a customer j $\in\{1,2\ldots N\}$ where $x_{0j}^r \neq 0$.

The driving time of a vehicle k is equal to time required to serve all customers of its routes. When the overtime for each vehicle is the additional time to time horizon that vehicle needs to perform its service. If the vehicle respects time constraint, there is no overtime and it is equal to 0. Overtime of a vehicle k is calculated in literature as follows [Brandao and Mercer, 1998]:

$$OT_k = max\left(0, \sum_{r\in S_k} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j\neq i}}^{N} t_{ij} x_{ij}^r - \quad T\right) \tag{2}$$

Overtime of the solution is the maximum overtime of vehicles. In other words, it is the overtime of the latest vehicle:

$$OT = \max_{0\le k\le K} OT_k \tag{3}$$

The mathematical model consists in minimizing the maximum overtime and routing cost as follows:

$$min\ (\max_{0\le k\le K} (max\left(0, \sum_{r\in S_k} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j\neq i}}^{N} t_{ij} x_{ij}^r - T\right))) \tag{4}$$

$$min \sum_{r=1}^{nK} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j\neq i}}^{N} d_{ij} x_{ij}^r \tag{5}$$

Under the following constraints:

$$\sum_{i=0}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} q_j x_{ij}^r \le Q \qquad r{:}\ 1\ldots nK \tag{6}$$

$$\sum_{r\in S_k} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j\neq i}}^{N} t_{ij} x_{ij}^r - T \le \alpha T \qquad k = 1 \ldots K \tag{7}$$

$$\sum_{r=1}^{nK}\sum_{\substack{i=0\\i\neq h}}^{N} x_{ih}^{r} = 1 \qquad h: 1\ldots N \tag{8}$$

$$\sum_{\substack{i=0\\i\neq h}}^{N} x_{ih}^{r} = \sum_{\substack{j=0\\j\neq h}}^{N} x_{hj}^{r} \qquad h: 0..N, r: 1\ldots nK \tag{9}$$

$$\sum_{i\in S}\sum_{\substack{j\in S\\j\neq i}} x_{ij}^{r} \leq |S| - 1 \qquad r: 1\ldots nK, S \subset \{1\ldots N\} \,/\, 2 \leq |S| \leq N \tag{10}$$

$$x_{ij}^{r} \in \{0,1\} \qquad r: 1\ldots nK \;, i: 0\ldots N, j: 0\ldots N \quad i \neq j \tag{11}$$

The first Objective in Eq. (4) minimizes the maximum overtime of all the vehicles of the solution, and the second objective in Eq. (5) aims at minimizing the routing cost. Capacity constraints are formulated with Eq. (6). As for Eq. (7), overtime is restricted to a maximum legal overtime. In Eq. (8), each customer is visited exactly once. Constraints in Eq. (9) force each vehicle that visits a customer to leave it after the delivery, and each vehicle that leaves the depot to return to the depot at the end of the trip. Finally, the creation of sub routes is prevented by Eq. (10).

VRPM is a generalization of VRP which is known to be NP-Hard. Thus, the proposed mathematical model cannot be solved in a polynomial time to find the best solution of the problem. It is used essentially to describe the problem specifications and the objective functions. Therefore, a meta-heuristic is used to solve this problem in order to find a good solution in a short time.

4. Memetic Algorithm

Memetic algorithm (MA) belongs to the larger class of evolutionary algorithms that use techniques inspired from natural evolution. It is a combination of genetic algorithm (GA), first introduced by [Holland, 1975], and local search procedures that intensify the search. GA is inspired from the biological evolution, whereas MA mimics cultural transmission based on the meme concept analogous to the gene in GA [Dawkins, 1976]. It is also called hybrid genetic algorithm or genetic local search algorithm. The memetic algorithm nomination was first introduced by Moscato [Moscato, 1989]. Since then, MA has been used in several works and has proved its efficiency to resolve VRP.

The proposed MA implements genetic operators as selection, mutation and crossover, and also develops some local search operators that improve the quality of the individuals in order to speed up the evolution of the population. This way, the algorithm combines benefits of the population based search approach that can do parallel search, with efficient heuristics that explore domain knowledge with linear search. The flowchart of the proposed algorithm is given in Fig. 2. The MA steps are detailed in the following paragraphs.

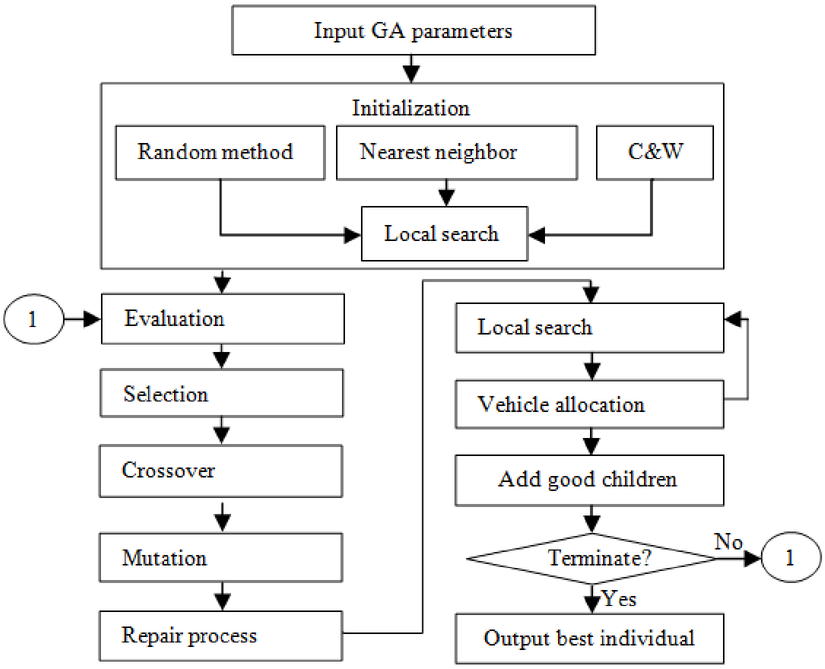

Fig. 2: The flowchart of memetic algorithm

4.1. *Solution Encoding*

For the problem at hand, the encoding of a candidate has to deal efficiently with the two level information of the multiple trip solution: Allocation of routes to vehicles, and the delivery order for each route. In the adopted representation, the chromosome (individual) contains several trips; the term "trip" is used for the genetic representation to designate a complete journey of a vehicle, it contains routes assigned to the same vehicle, separated by a delimiter 0. Each route is composed by an ordered subset of customers.

The main advantage of the proposed encoding is that it is a natural representation and describes all features of the real solution. Consequently, we can switch easily from the chromosome to the solution and vice versa. As an example, the chromosome from Fig. 3 represents the solution presented above in Fig. 1.

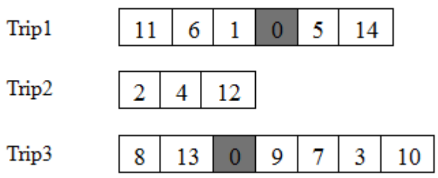

Fig. 3: Encoding of solution of Fig.1

4.2. *Population initialization*

To accelerate algorithmic convergence, it is advisable to use a fast and simple heuristic to find a good solution for the first generation. This approach can significantly reduce GA's computational time required to reach a reasonable local minima. For this purpose, the algorithm use two good heuristics; the savings procedure of [Clark & Wright, 1964] that was frequently used by many researchers and gave satisfactory results, and the second algorithm is the neighbor search algorithm which is one of the most known heuristics for VRP.

GA needs also to explore widely the search space in order to avoid falling into a local optimum that can be caused by an initial solution of high quality. For this reason, a random insertion method is proposed to find further individuals for the initial population. Each solution generated for the initial population is exposed for improvement attempts using the local search operators that will be presented later.

4.2.1. Saving method

The saving method proposed uses the sequential version of Clarck & Wright (C&W) algorithm where exactly one route is built at each iteration. It starts with an initial solution in which each customer is served individually. Then, the saving cost is calculated, for each pair of routes, by joining the two routes together as long as the resulting route respects capacity constraint and the maximal time of work (horizon time + legal overtime). The routes of the best saving are combined, and a new iteration begins to re-calculate savings and merge new routes. The algorithm is stopped if no merge, respecting the problem constraints, can be done.

The saving obtained from joining two routes R and S equals the saving of joining the last node of R say i, with the first node of S say j. The saving S_{ij} is calculated using the following formula, where d_{ij} refers to the routing cost between i and j.

$$S_{ij} = d_{0i} + d_{j0} - d_{ij} \tag{12}$$

4.2.2. Nearest neighbor method

The nearest neighbor method is easy to implement and is not time-consuming. The nearest customer to depot is selected first and added to the route. Then, the nearest non visited customer to the last one added is selected and inserted in the queue of the route, and so on until the maximal capacity or time are reached. A new route is then created and the insertion of the remaining customers is carried out in the same way.

4.2.3. Random insertion method

The random insertion method is a sequential insertion based on a random selection of customers. An initial route is created and customers are added, one by one, in a random order. A new route is created when adding a new selected customer exceeds the vehicle capacity or working time. The insertion is stopped if all customers are added.

4.3. *Evaluation*

The objective is processed hierarchically: First, the overtime is minimized until finding a solution with no overtime. Then, routing cost is minimized. In first step, each chromosome is measured by an evaluation function that minimizes maximum overtime, in order to find a feasible solution that respects time horizon. When a feasible solution is reached, the first objective is no longer useful. Then, the algorithm proceeds to search other feasible solutions with a better routing cost.

The evaluation function is calculated for individuals so that they can be compared to each other in order to determine which solutions get passed onto the next generation of population. Individuals are compared as follows: If one or two of them exceed time horizon, the best individual is one that has lower overtime. If both respect time constraint, the individual that has the minimal routing cost is considered the best one.

4.4. *Crossover*

A crossover operator is a genetic operator that combines two chromosomes (parents) to produce a new chromosome (child). The idea behind the crossover operator is that the new chromosome may be better than both of the parents if it takes the best characteristics of each one of them. It is applied with a crossover probability p_c.

During each successive generation, two individuals of the existing population are selected to breed a new generation. The crossover operator requires some strategy to select parents from the previous generation. In this work, the two parents are randomly selected from the population.

The crossover operator used in our approach does not promote a mutual exchange of genetic material between two parents in the same operation. In the first step, one parent is selected to be recipient and the other one is a donor, the recipient parent is submitted to a crossover operation to generate the first child. Then, roles are reversed to generate the second child. The crossover operates in the following way: the longest trip of the recipient parent that causes the highest overtime is removed. Missing customers are then copied from the donor parent in their visiting order within donor routes. The example from Fig. 4 illustrates how crossover acts.

4.5. *Mutation*

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population to the next. It avoids the algorithm to have local minima by preventing the population of chromosomes from becoming too similar.

Two random operators are used in the mutation phase of the proposed MA in order to input new characteristics to the current population and enlarge the search space. (Fig. 5)

- Random insertion: this operator randomly chooses two routes. Then, a customer is chosen from one route and inserted into the other in a randomly selected position.
- Random exchange: this operator randomly chooses two customers from two different routes and exchanges them.

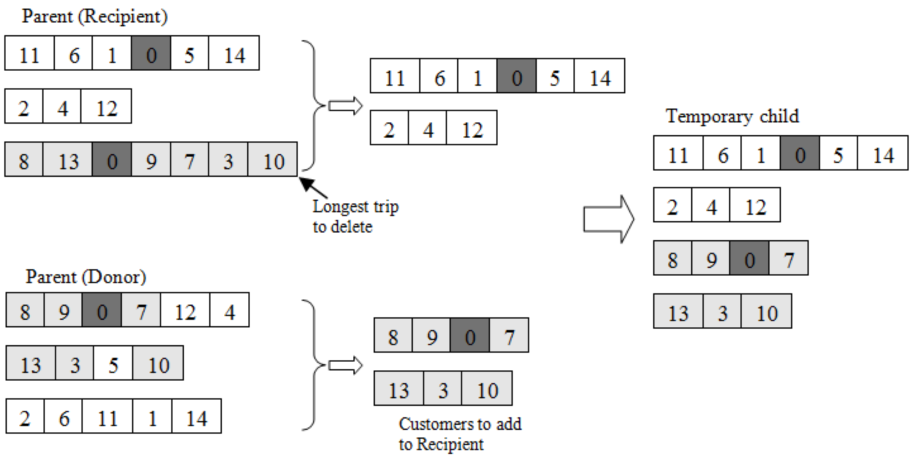

Fig. 4: A crossover example

Mutation is applied, with a mutation probability p_m, to chromosomes generated by the crossover operators, or directly to selected individuals if no crossover has been done. The mutation phase is followed later by a local search phase that uses also exchange and insertion moves. Thus, local search may turn the chromosome back to its initial state, essentially if no crossover was performed before. To avoid this situation, we choose to apply, during the mutation phase, both insertion and exchange operators for a predefined number of iterations. This helps also to be far enough away from local optima.

4.6. *Repair process*

After crossover and mutation operations, constraints are no longer respected in most cases. A repair process launches a local search for each route to improve the visit order of customers as will be described in 4.7.1. Afterwards, routes that exceed vehicle quantity or maximum travel time are splitted using partitioning position that ensures a minimal total routing cost of the two generated routes. Thereafter, short routes are combined using the C&W method already described in 4.2.1. Fig. 5 illustrates a repair process applied to chromosome generated by crossover and mutation in Fig. 3 and Fig. 4

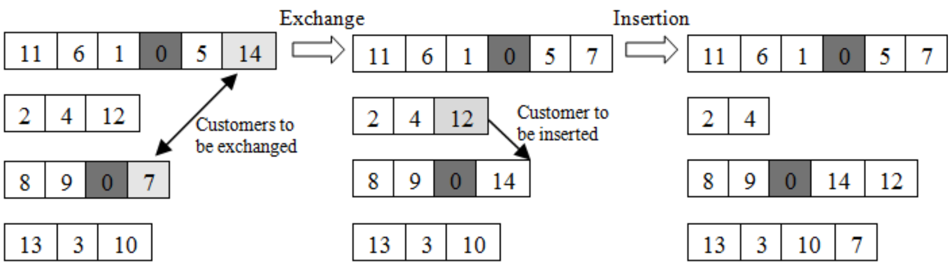

Fig. 5: Exchange and insertion mutations applied to child generated from crossover

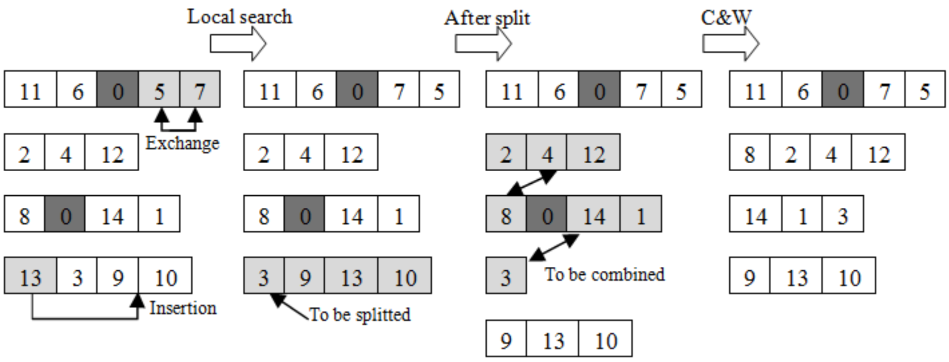

Fig. 6: Repair process applied to mutated child

4.7. *Local search*

For each solution generated for the initial population and for each new child obtained after crossover and mutation operators, we use a local improvement procedure. The proposed local search uses mainly two neighborhood moves classically used for VRPs: re-insertion and exchange. In order to speed up the search, we execute the first improvement move found rather than the best improvement move.

The procedure works as follow; for a selected individual, each route is improved separately with a route local search. Afterwards, a local search is applied between routes of the solution. Both route local search and solution local search are detailed in the following paragraphs.

4.7.1. Route local search

Route local search considers each route in isolation. In the first step, the cross remover operator checks if there is any cross between two arcs of the considered route and removes it as illustrated in Fig. 7. Then, 2-opt swaps are tried between customers in order to minimize routing cost and consequently travel time, until there is no possible exchange. Afterwards, the re-insertion operator chooses a customer and moves it to the best position that ensures a minimal travel time/distance of the route. The operation is repeated for all customers of the route. (Fig. 7)

4.7.2. Solution local search

In this step of the local search procedure, 2-opt and re-insertion operators are applied between two different routes of the solution (Fig. 8). First of all, routes are sorted from longest to shortest in terms of time/distance. In fact, the longest routes are the main responsible for overtime and the high routing cost, so they are the first to be improved.

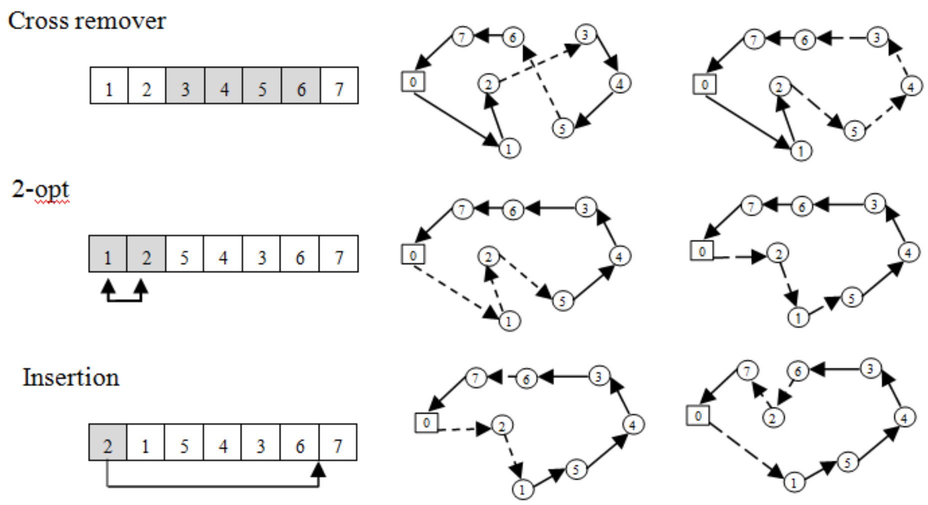

Fig. 7: Route local search

The procedure of the proposed local search is as follows:

(1) While the maximal number of moves is not reached

(i) Sort routes form longest to shortest $\{R_0, R_1, R_2 \ldots R_p\}$

(ii) For each route R_i from $\{R_1, R_2 \ldots R_p\}$

i. For each route R_j from $\{R_{i+1}, R_{i+2} \ldots R_p\}$

1.ii.i.1 For each pair of customers c_i of R_i and c_j of R_j

1.ii.i.1.1 Try to exchange c_i and c_j if there is no violation constraint, and the total routing cost of R_i and R_j is reduced.

1.ii.i.1.2 Re-assign routes to vehicles and evaluate the new solution

1.ii.i.1.3 Execute the exchange if the new solution is better than the initial one

1.ii.i.1.4 Return to (1)

ii. For each customer ci of Ri,

1.ii.ii.1 If total quantity of c_i and R_j does not exceed vehicle capacity

1.ii.ii.1.1 Try to insert c_i into the best position of R_j.

1.ii.ii.1.2 Re-assign routes to vehicles and evaluate the new solution

1.ii.ii.1.3 Execute the move if the new solution is better than the initial one

1.ii.ii.1.4 Return to (1)

iii. No further improvement is possible, so stop the search

4.8. *Vehicle allocation*

Each trip (line) of the chromosome is assigned to one vehicle. In case the number of trips exceeds the number of vehicles, routes of the non assigned trip of the chromosome are moved to another trip.

In order to optimize the assignment of routes to vehicles, and to ensure a minimal overtime, routes are submitted to a number of moves from vehicle to another. Two routes

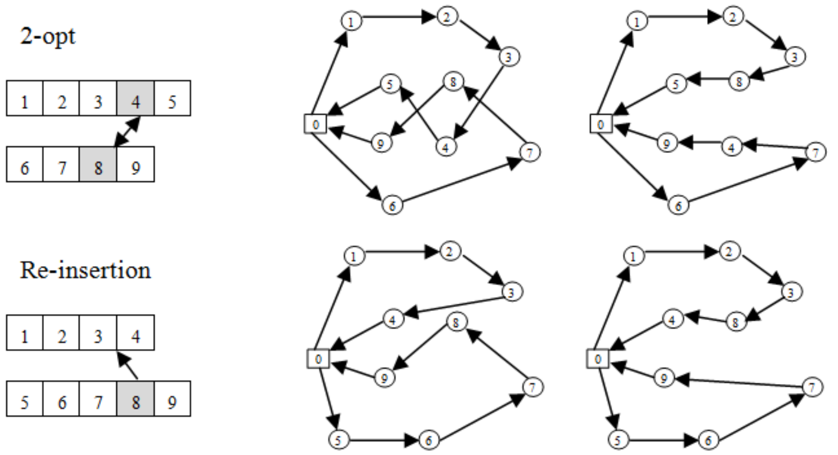

Fig. 8: Exchange and Re-insertion operators between two routes

selected from two different vehicles are to be exchanged if the maximal driver time of these two vehicles becomes lower after the exchange. The same condition authorizes a route to move from one vehicle trip to another. An example is illustrated in Fig. 9.

4.9. *Memetic parameters*

The memetic parameters are very important for the success of a memetic algorithm, as well as for the representation of the individuals, the initial population and the genetic operators. Based on preliminary computational experiments, we set the following genetic parameters:

- The population size is set to 100 individuals. This number was set for data sets of 50 to 199 customers, and is enough to guarantee a large search space.
- The crossover probability is set to 0.9 and the mutation probability is set to 0.4, those probabilities ensure a good intensification and diversification of the population.
- The number of iterations for the mutation is set to 3.
- The maximum number of generations in our algorithm is set to 6000.
- In this algorithm, the termination strategy is based on a maximum number of generations with no improvement in the objective function. In the first step of

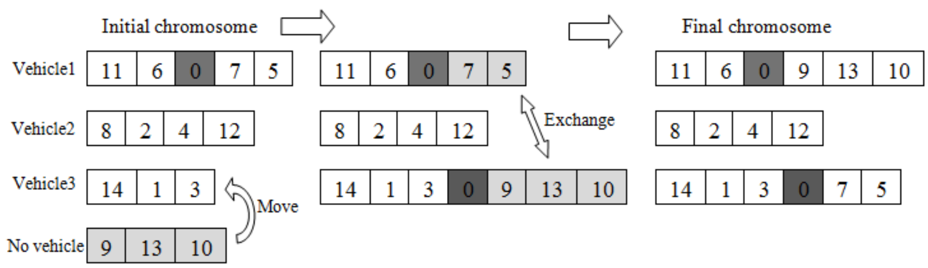

Fig. 9: Vehicle allocation

minimizing overtime, this number is fixed to 2000, in order to intensify the search for a feasible solution. When a solution is found, the algorithm tries to improve routing cost and stops after a sequence of 500 consecutive generations without improvement.

- Local search is executed until no move is possible or the maximum number of iterations set to 50 is reached.

5. Computational results

The test problems for VRPM have been suggested by [Taillard *et al.*., 1996]. They use a nine base problem (PB); seven test problems (C) from [Christofides *et al.*, 1979] and two test problems (F) from [Fisher, 1994]. They have used the same graphs, demands and vehicle capacities of those problems, and have generated several instances using different values of m (number of vehicles allowed), and two values of time horizon T_1=[1,05 z^*/m] and T_2=[1,1 z^*/m], where z* is the value of a VRP solution obtained in [Rochat and Taillard, 1995].

The proposed algorithm has been coded in Java and executed on an Intel core i3 processor at 2.4 GHz with 4Go of RAM under Windows 7. The performance of the suggested algorithm is verified by a comparison to benchmarks dealing with similar problem using the same data sets; [Taillard, Laporte and Gendreau, 1996], [Brandao and Mercer, 1998], [Patch and Salhi, 2004], [Olivera and Viera, 2007], [Salhi and Petch, 2007], [Alonso *et al*, 2008] and [Mingozzi *et al.*, 2012]. From now on, these benchmarks will be referred to as TLG96, BM98, PS04, OV07, SP07, AAB08 and MRT12 respectively, and our algorithm will be named MA. For all the data sets, time and distance are considered to be equivalent. For each instance, we performed just one run using memetic algorithm.

In this paper, we present the results for the whole benchmark. Algorithm has been run completely over instances of 9 problems (*C1*, *C2*, *C3*, *C4*, *C5*, *C11*, *C12*, *F11* and *F12*). In total, 104 instances are used for computational tests. AAB08 approach has been tested only for the 92 instances of C problems and MRT12 deal with 56 instances of five problems (*C1, C2, C3, C11* and *C12*). For the other works, the 104 benchmark instances have been used.

The subject of this computational execution is to assess the quality of our algorithm. For this purpose, we need to compare our results with those of previous works. The comparison should include the two objectives: minimization of the maximum overtime for infeasible solutions, and minimization of the routing cost for feasible ones.

5.1. *Infeasible solutions*

A solution is called infeasible if it does not respect the horizon time. For all the studied benchmarks, infeasible solutions are qualified by the Longest Trip Rate (LTR) that compares time of the longest trip of a solution to the time horizon $T = T1$ or $T2$. For infeasible solutions, the value of LTR is higher than 1.

$$LTR = \frac{\max_{k \in \{1..K\}} t(k)}{T} \quad (13)$$

Where *t(k)* is the time required by vehicle *k* to serve all his routes.

Table 1 gives the number of infeasible solutions reported by each of the seven approaches. The proposed MA found 94 feasible solutions among 104 instances. For each of the 52 instances with *T2*, MA was able to reach a feasible solution, as well as BM98, OV07 and AAB08 that solve all the *T2* instances. For the instances with shorter time horizon *T1*, finding a feasible solution is more difficult. MA found 42 feasible solutions out of 52. For the 10 unsolved instances, no feasible solution was found by previous approaches except OV07 that solved 4 of them, and MRT12 that solve 2 of them using an exact method.

Note that the MA results reported below are found after only one run, whereas OV07 execute five runs for each instance. Further executions for those ten instances with MA may help to find feasible solutions or some infeasible ones with better LTR.

Table 1. The number of infeasible solutions found by MA and other benchmarks

PB	Z*	size	# inst	MA	TLG96	BM98	PS04	OV07	SP07	AAB08	MRT12
C1	524.61	50	8	2	3	2	4	2	3	2	2
C2	835.26	75	14	1	3	2	2	1	6	2	1
C3	826.14	100	12	1	4	1	2	0	4	1	0
C4	1028.42	150	16	2	4	2	3	1	8	2	-
C5	1291.44	199	20	1	1	2	6	0	11	2	-
C11	1042.11	120	10	1	1	1	2	0	5	1	0
C12	819.56	100	12	1	4	3	1	1	2	2	1
F71	241.97	71	6	1	3	2	2	1	3	-	-
F134	1162.96	134	6	0	0	0	0	0	0	-	-
Total			104	10	23	15	22	6	42	12/92	4/56

The longest trip rate of the ten infeasible solutions is presented in table 2. Bold LTR values indicate the best one among all the reported results.

Table 2. LTR comparison for infeasible solutions found by MA

PB	T	m	MA	TLG96	BM98	PS04	OV07	SP07	AAB08	MRT12	Dev
C1	184	3	**1,023**	1,115	1,041	1,026	1,024	1,03	1,041	≥1	-0,001
C1	138	4	**1,027**	**1,027**	**1,027**	1,085	**1,027**	1,056	**1,027**	≥1	0
C2	125	7	1,034	1,073	1,088	1,064	**1,009**	1,102	1,05	≥1	0,025
C3	145	6	1,008	1,032	1,003	1,001	**≤1**	1,05	1,002	**≤1**	0,008
C4	154	7	1,024	1,033	1,071	1,072	**1,002**	1,09	1,041	-	0,022
C4	134	8	1,016	1,075	1,031	1,058	**≤1**	1,1	1,038	-	0,016
C5	136	10	1,003	1,024	1,051	1,064	**≤1**	1,076	1,045	-	0,003
C11	274	4	1,001	1,02	1,011	1,052	**≤1**	1,052	1,038	**≤1**	0,001
C12	143	6	**1,014**	1,064	1,072	1,029	**1,014**	1,029	1,113	≥1	0
F71	85	3	1,02	1,075	**1,011**	1,02	1,02	1,02	-	-	0,009

Deviation to the best known LTR is the difference between the LTR found by MA and the best LTR found by the other works. It is calculated this way:

$$Dev = LTR_{MA} - max(\min_{1\leq p\leq 7} LTR_p \,,1) \quad (14)$$

LTR_p is the LTR found by an approach p of the seven benchmarks studied.

Infeasible solutions obtained are of high quality, as the maximal *LTR* found is equal to 1,034 i.e. the worst overtime is about 3,4% of the time horizon. The algorithm obtains one new best solution for the instance *C1* with *m*=3 and *T*=184, and riches the best *LTR* found by the other benchmarks in two instances. For the other instances, deviation from the best *LTR* is very small; it does not exceed 0.01 in three cases, and 0.025 in the other cases.

5.2. *Feasible solutions*

For feasible solution, the time horizon is respected and the objective here is to minimize the routing cost. The cost of feasible solutions was reported only by SP07 and MRT12. OV07 approach gives only the *GAP* of the cost to z^* the best known solution of VRP. While the other authors mention the cost just for the infeasible solutions found by their algorithm. Therefore, comparison of the routing cost will be made with SP07, OV07 and MRT12 using the *GAP* defined by OV07.

The *GAP* of a solution s measures its quality by comparing its routing cost value to z^*. Let *f(s)* be the cost of the solution s, the *GAP* is formulated as follows:

$$GAP(s) = 100 * \left(\frac{f(s)}{z^*} - 1\right) \quad (15)$$

OV07 benchmark provides the best, average and worst *GAP* values for feasible solutions found considering five runs. The *GAP* found by MA will be compared to the best one reported by OV07. In order to compare also with SP07 and MRT12, we calculate the *GAP* for their results using costs reported in their papers. Note that OV07 gives *GAP* rounded to 1 decimal place, and we use 2 decimals for a better precision. When the *GAP* values need to be compared to OV07, they are rounded to 1 decimal. The routing cost is compared between two approaches when a feasible solution is found by both of them.

Detailed results are given in Table A.1 in the appendix A. Table 3 shows a synthesis on the feasible solutions quality found by MA, OV07, SP07 and MRT12. It presents the average *GAP* and the worst one for each approach, the number of solutions that reach z^* (*GAP*=0) and solutions too near to z^* (*GAP* $\leq$ 0.1). It reports also the number of solutions that reach optimal solutions found by MRT12 ($GAP=GAP_{MRT}$), the number of best known solutions found by each approach (Best) and precisely the number of best known solutions provided only by the approach (Strictly best).

In term of routing cost, MA produces feasible solutions of good quality. The average GAP is around 1.47. For the 94 feasible solutions found, MA reaches z* in 14 instances and find the optimal solution z_{MRT} in 13 instances. In 9 cases, solutions found are very close to z* with a GAP value less than 0.1.

Comparing with SP07, all results found by MA are strictly better than those found by SP07, except for one where the difference between GAPs is about 0.3 (Table A.1). MA reaches the best routing cost in 38 cases; it has been able to provide new best feasible solutions for 22 instances and produces equal quality results to best solutions proposed by OV07 or MRT12 in 16 instances.

OV07 reaches z* for 16 cases and z_{MRT} for 18 cases, with a better average *GAP* of 1.28. It finds the best solution in 51 instances including the four feasible solutions that have not been found by MA. For 28 instances, the best known solution has been found only by OV07.

The approach that give the best results is the exact algorithm of MRT12 that solve to optimality 42 out of 52 benchmark instances with up to 120 customers. The average GAP is equal to 0.7.

Table 3. Synthesis on the feasible solutions quality.

	# Feasible Solutions	Average GAP	worst GAP	# GAP = 0	# GAP ≤ 0,1	# GAP = GAP_{MRT}	#Best	#Strictly Best
MA	94	1,47	6,89	14	23	13	38	22
SP07	62	4,53	9,27	0	2	0	0	0
OV07	98	1,28	5,4	16	27	18	51	30
MRT12	52	0,7	5,35	35	1	42	42	18

5.3. *Computing time*

During its execution, the algorithm takes much time to return a result for big instances. It can reach up to 4 hours in some cases. Note that in most cases, high running time is not due to searching feasible solution but rather to searching the best routing cost among feasible solutions. For a considerable number of instances, the saving heuristic used for the initialization of memetic algorithm combined with local search heuristic led a feasible solution in a very short time. However, searching the shortest feasible solution in term of cost can slow down the algorithm. MA needs some refinements especially for the local search algorithm to speed up its execution. Table 4. presents the average of CPU time for each of the nine problems.

Table 4. The average of execution time for each problem.

Problem	Time(s)	Time(min)
C1	43	0,72
C2	365	6,08
C3	3751	62,52
C4	5810,58	96,84
C5	7040,69	117,34
C11	2347,67	39,13
C12	0,54	0,01
F71	3680,89	61,35
F134	12632,6	210,54

6. CONCLUSION

In this work, we have dealt with an important extension of Vehicle Routing problem, namely the Multi-Objective Vehicle Routing Problem with Multiple Trips. A mathematical model has been formulated and a memetic algorithm has been proposed to solve the problem. We have conducted computational experiments on a set of 104 benchmark instances from the literature. The algorithm has been compared with seven existing approaches. It has proved to be much more efficient than five heuristics described in the literature, and has shown competitive results to one heuristic proposed by [Olivera *et al.*, 2007]. Our approach solves to optimality 13 instances out of 42 optimal solutions provided by [Mingozzi *et al.*, 2012] using an exact method. It has contributed 23 new best solutions for the 104-instances tested. However, the proposed algorithm is very expensive in terms of computing time, hence the need of some refinements for the local search procedure to speed up the execution. As a future work, the proposed algorithm can be applied to further variants of VRPM closer to real world applications like considering heterogeneous fleet.

Appendix A. Appendix

Table A.1. provides detailed computational results found by MA. It reports the finishing time of the latest vehicle (time), the routing cost, LTR and the GAP to z* for MA approach and the GAP for OV07, SP07 and MRT12.

Bold values indicate the best one among all the reported results.

Table A.1. Results for one run over the benchmark problem set.

			MA				**OV07**	**SP07**	**MRT12**
BP	**T**	**NV**	**time**	**Cost**	**LTR**	**GAP**	**GAP**	**GAP**	**GAP**
CMT1	551	1	524,92	524,92	0,953	0,06	**0,0**	4,13	**0,00**
	275	2	274,49	536,56	0,998	2,28	**1,6**	-	**1,60**
z*=524,61	184	3	188,24	561,01	1,023	-	-	-	-
	138	4	141,76	547,10	1,027	-	-	-	-
	577	1	524,94	524,94	0,910	0,06	**0,0**	4,29	**0,00**
	289	2	288,90	530,79	1,000	1,18	**1,0**	4,73	**1,00**
	192	3	190,72	556,61	0,993	6,10	**5,4**	6,80	**-**
	144	4	141,79	546,43	0,985	4,16	**4,1**	8,05	**4,13**
CMT2	877	1	835,77	835,77	0,953	0,06	**0,0**	4,05	**0,00**
	439	2	419,85	839,58	0,956	0,52	0,2	3,62	**0,00**
z*=835,26	292	3	281,46	836,18	0,964	0,11	0,2	-	**0,00**
	219	4	213,27	835,77	0,974	0,06	0,1	2,58	**0,00**
	175	5	174,32	839,71	0,996	0,53	1,2	-	**0,06**
	146	6	145,92	861,88	0,999	3,19	**2,6**	-	-
	125	7	129,26	885,57	1,034	-	-	-	-
	919	1	835,26	835,26	0,909	**0,00**	0,1	4,13	**0,00**
	459	2	418,66	835,26	0,912	**0,00**	0,1	5,54	**0,00**
	306	3	280,16	835,26	0,916	**0,00**	0,1	4,05	**0,00**
	230	4	220,54	835,77	0,959	0,06	0,1	5,46	**0,00**
	184	5	178,7	835,77	0,971	0,06	0,1	5,75	**0,00**
	153	6	152,57	842,28	0,997	0,84	2,5	-	**0,47**
	131	7	130,71	870,19	0,998	4,18	**3,6**	-	-
CMT3	867	1	830,00	830,00	0,957	0,47	**0,0**	2,32	**0,00**
	434	2	421,09	828,74	0,970	0,31	0,4	2,97	**0,00**
z*=826,14	289	3	280,40	829,91	0,970	0,46	0,5	-	**0,00**
	217	4	213,06	828,74	0,982	**0,31**	0,4	-	**-**
	173	5	172,23	833,98	0,996	**0,95**	2,6	-	-

Table A.1. (Continued) Results for one run over the benchmark problem set.

BP	T	NV	MA				OV07	SP07	MRT12
			time	Cost	LTR	GAP	GAP	GAP	GAP
	145	6	146,14	867,72	1,008	-	**1,2**	-	-
	909	1	828,26	828,26	0,911	0,26	0,4	2,32	**0,00**
	454	2	414,55	828,26	0,913	0,26	0,3	5,56	**0,00**
	303	3	282,14	829,51	0,931	0,41	0,2	5,25	**0,00**
	227	4	219,99	829,54	0,969	0,41	**0,0**	6,28	**0,00**
	182	5	174,48	834,42	0,959	1,00	**0,8**	9,10	-
	151	6	150,14	836,56	0,994	1,26	**1,1**	4,31	-
CMT4	1080	1	1048,98	1048,98	0,971	2,04	**0,5**	3,51	-
	540	2	520,90	1041,09	0,965	1,27	**0,8**	3,68	-
z*=1028	360	3	351,09	1038,42	0,975	1,01	**0,7**	-	-
	270	4	268,07	1053,20	0,993	2,45	**0,8**	-	-
	216	5	211,19	1039,08	0,978	1,08	**0,4**	-	-
	180	6	179,47	1054,19	0,997	2,55	**2,0**	-	-
	154	7	157,62	1062,71	1,024	**-**	-	-	-
	134	8	136,08	1066,97	1,016	-	**3,0**	-	-
	1131	1	1075,95	1075,95	0,951	4,66	**1,0**	5,93	-
	566	2	524,070	1044,69	0,926	1,62	**0,9**	4,13	-
	377	3	345,46	1034,60	0,916	0,64	**0,5**	4,79	-
	283	4	179,47	1054,19	0,634	2,55	**1,0**	8,86	-
	226	5	224,55	1045,37	0,994	1,69	**0,8**	5,58	-
	189	6	181,25	1049,78	0,959	2,12	**0,4**	8,17	-
	162	7	161,44	1055,04	0,997	**2,63**	**2,6**	-	-
	141	8	140,00	1057,52	0,993	**2,87**	3,5	-	-
CMT5	1356	1	1313,2	1313,22	0,968	**1,69**	1,9	4,33	-
	678	2	657,03	1312,58	0,969	**1,64**	2,0	4,27	-
z*=1291,44	452	3	437,93	1313,21	0,969	1,69	**1,6**	-	-
	339	4	337,11	1334,02	0,994	3,30	**2,5**	-	-
	271	5	268,42	1322,87	0,990	**2,43**	2,5	-	-
	226	6	223,24	1323,64	0,988	**2,49**	3,1	-	-
	194	7	193,21	1330,96	0,996	**3,06**	3,6	-	-
	170	8	169,60	1338,50	0,998	3,64	**2,8**	-	-
	151	9	150,70	1324,05	1,00	**2,53**	3,4	-	-
	136	10	136,45	1346,09	1,00	4,23	**4,1**	-	-
	1421	1	1316,5	1316,48	0,926	**1,94**	2,1	-	-
	710	2	690,2	1380,38	0,972	6,89	**1,8**	8,38	-
	474	3	459,9	1378,68	0,970	6,76	**1,6**	9,13	-
	355	4	330,94	1319,59	0,932	2,18	**1,3**	8,22	-
	284	5	270,70	1340,45	0,953	3,79	**2,0**	9,27	-
	237	6	224,59	1324,65	0,948	2,57	**1,5**	6,63	-
	203	7	192,16	1326,85	0,947	2,74	**2,3**	8,00	-
	178	8	169,89	1306,68	0,95	**1,18**	1,8	-	-
	158	9	152,19	1327,19	0,96	**2,77**	3,0	-	-
	142	10	140,57	1316,40	0,99	**1,93**	4,1	-	-
CMT11	1094	1	1042,11	1042,11	0,95	**0,00**	0,1	4,43	**0,00**
	547	2	542,14	1047,00	0,991	0,47	3,0	-	**0,00**
z*=1042,11	365	3	353,21	1054,17	0,968	1,16	0,2	-	**0,00**
	274	4	274,31	1079,17	1,001	-	**4,2**	-	-
	219	5	215,15	1046,34	0,982	0,41	0,1	-	**0,00**
	1146	1	1047,62	1047,62	0,914	0,53	0,2	4,43	**0,00**
	573	2	536,42	1049,51	0,936	0,71	1,2	6,52	**0,00**
	382	3	280,62	1048,58	0,735	0,62	0,1	4,46	**0,00**
	287	4	279,39	1042,11	0,973	**0,00**	0,1	-	**0,00**
	229	5	213,63	1042,11	0,933	**0,00**	0,1	4,88	**0,00**

Table A.1. (Continued) Results for one run over the benchmark problem set.

BP	T	NV	MA time	MA Cost	MA LTR	MA GAP	OV07 GAP	SP07 GAP	MRT12 GAP
CMT12	861	1	819,56	819,56	0,952	**0,00**	**0,0**	0,05	**0,00**
	430	2	413,20	819,56	0,961	**0,00**	**0,0**	0,22	**0,00**
z*=819,56	287	3	276,55	819,56	0,964	**0,00**	**0,0**	0,91	**0,00**
	215	4	212,89	824,32	0,990	0,58	**0,0**	0,61	**0,00**
	172	5	172,00	842,08	1,000	**2,75**	3,1	-	-
	143	6	144,99	855,25	1,014	-	-	-	-
	902	1	819,56	819,56	0,909	**0,00**	**0,0**	0,05	**0,00**
	451	2	410,19	819,56	0,910	**0,00**	**0,0**	1,22	**0,00**
	301	3	276,55	819,56	0,919	**0,00**	**0,0**	3,86	**0,00**
	225	4	212,89	824,32	0,946	0,58	**0,0**	0,24	**0,00**
	180	5	178,84	824,78	0,994	**0,64**	**0,6**	1,74	**0,64**
	150	6	148,17	823,15	0,988	**0,44**	**0,4**	4,37	**0,44**
F71	254	1	241,97	241,97	0,953	**0,00**	**0,0**	5,06	-
	127	2	126,81	250,85	0,999	**3,67**	4,2	-	-
z*=241,97	85	3	86,66	258,53	1,020	-	-	-	-
	266	1	241,97	241,97	0,910	**0,00**	**0,0**	5,00	-
	133	2	132,97	248,58	1,000	2,73	**0,0**	5,00	-
	89	3	87,68	254,87	0,985	**5,33**	5,4	6,02	-
F134	1221	1	1166,96	1166,96	0,956	**0,34**	0,5	2,34	-
	611	2	582,84	1163,53	0,954	**0,05**	0,7	2,69	-
z*=1162,96	407	3	392,39	1162,97	0,964	**≈0,00**	0,3	3,17	-
	1279	1	1166,71	1166,71	0,912	**0,32**	0,8	1,72	-
	640	2	582,84	1163,53	0,911	**0,05**	0,7	3,15	-
	426	3	409,70	1165,98	0,962	**0,26**	0,4	4,51	-

References

Alonso, F., Alvarez, M.J. and Beasley, J.E. (2008). A Tabu Search Algorithm for the Periodic Vehicle Routing Problem with Multiple Vehicle Trips and Accessibility Restrictions, Journal of the Operational Research Society, 59, 963-976

Azi, N., Gendreau, M. and Potvin, J-Y. (2007). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. European Journal of Operational Research, 178:755-766.

Azi, N., Gendreau, M. and Potvin, J-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. European Journal of Operational Research, 202:756-763.

Azi, N., Gendreau, M. and Potvin, J-Y. (2011). A dynamic vehicle routing problem with multiple delivery routes. Annals of Operations Research, In-press.

Battarraa, M., Monacib, M. and Vigoa D. (2009). An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem, Computers & Operations Research 36 3041-3050

Brandao, J. and Mercer, A. (1997), A Tabu Search Algorithm for the Multi-trip Vehicle Routing and Scheduling Problem, European Journal of Operational Research, 100, 180-191

Brandao, J. and Mercer, A. (1998), The Multi-trip Vehicle Routing Problem, Journal of the Operational Research Society, 49, 799-805

Christofides N., Mingozzi A. and Toth P. (1979). The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P and Sandi C (eds). Combinatorial Optimization. Wiley, Chichester, pp 315-338.

Clarke, G., and Wright, J.W. (1964), "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research 12, 568-581.

Dantzig, G. B. and Ramser, J. H., (1959). The truck dispatching problem, Management Science, Vol. 6; pp 80-91.

Dawkins, R., (1976). The Selfish Gene. Oxford University Press. ISBN 0-19-929115-2.

Derigs, U., Kurowsky, R., Vogel, U. (2011). Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services, European Journal of Operational Research 213 309-319

Eksioglu B., Vural A. V. and Reisman A. (2009). The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering, Vol. 57 (4), pp. 1472-1483.

Fisher, M. (1994), Optimal Solution of Vehicle Routing Problems Using Mimimum k-trees. Operations Research, 42-4, 626-646

Fleischmann, B. (1990), The Vehicle Routing Problem with Multiple Use of Vehicles. Working paper, FacbereichWirtschaftswissenschafteUniversitat Hamburg.

Golden, B.L.,Laporte, G. and Taillard, E.D. (1996), An Adaptive Memory Heuristic for a Class of Vehicle Routing Problems with Minmax Objective, Compurters and Operations Research, 24-5, 445-452

Hernandez, F., Feillet, D., Giroudeau, R. and Naud, O. (2010). An exact method to solve the multi-trip vehicle routing problem with time windows and limited duration. In TRISTAN VII, pages 366-369.

Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

Huang, C. and Lee, CM. (2011). A Study of Multi-Trip Vehicle Routing Problem and Distribution Center Location Problem Institute of Industrial Engineering and Management.POMS 22nd Annual Conference, Reno, Nevada, U.S.A. (April 29 to May 02, 2011)

Lin, C.K.Y., Kwok, R.C.W. (2006). Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. European Journal of Operational Research 175 1833–1849

Macedo, R., Alves, C., Valério de Carvalho, J.M.,Clautiaux, F., Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model, European Journal of Operational Research 214 536–545

Moscato, P. (1989). On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms". Caltech Concurrent Computation Program (report 826).

Mingozzi, A., Roberti, R., and Toth, P. (2012): An Exact Algorithm for the multi-trip vehicle routing problem. INFORMS Journal on Computing, Articles in Advance, pp. 1-15

Olivera, A. and Viera, O. (2007).Adaptive Memory Programming for the Vehicle Routing Problem with Multiple Trips. Computers and Operations Research, 34, 28-47

Petch, R.J. and Salhi, S. (2004). A Multi-phase Constructive Heuristic for the Vehicle Routing Problems with Multiple Trips. Discrete Applied Mathematics, 133, 69-92

Salhi. S. andPetch, R.J. (2007): A GA Based Heuristic for the Vehicle Routing Problem with Multiple Trips. J. Math. Model. Algorithms 6(4): 591-613.

Prins C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multi trip VRP with Application to a Large-Scale Real Case. Journal of Mathematical Modeling and Algorithms 1: 135-150.

Rochat Y. and Taillard E. (1995). Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Publication CRT-95–13, Centre de recherche sur les transports, Montreal.

Salhi, S. and Petch, R.J. (2007). A GA Based Heurisric for the Vehicle Routing Problem with Multiple Trips, Journal of Mathematical Modeling and Algorithms, 6-4, 591-613

Şen, A. and Bülbül, K.A (2008). Survey on multi trip vehicle routing problem. International Logistics and Supply Chain Congress (November 6-7, 2008), Istanbul, TURKIYE

Taillard, E., Laporte, G. and Gendreau, (1996). Vehicle Routing Problem with Multiple Use of Vehicles. Journal of the Operational Research Society, 47, 1065-1070

Toth, P. and Vigo,D. (2001). The Vehicle Routing Problem, SIAM.