Evolutionary Local Search Algorithm to Solve the Multi-Compartment Vehicle Routing Problem with Time Windows

Jan Melechovský[1]

Abstract. Vehicle routing problems (VRPs) represent an important research stream in applied combinatorial optimization. As the problems emerging from the practice become more constrained and large, more sophisticated approaches are required to address them. This paper tackles the Multiple Compartment Vehicle Routing Problem with Time Windows. This extension of the classical VRP consists of an unlimited homogeneous fleet of vehicles, each being equipped with multiple compartments. Such a configuration enables to load distinct commodities. A customer is characterized by a nonnegative demand of each commodity and a time window allowing its delivery. The goal is to service all customers under the vehicle capacity constraints and time windows constraints with minimal total cost. A metaheuristic solution approach based on evolutionary local search is presented. The performance of the algorithm is evaluated on benchmarks available in the literature.

Keywords: multiple compartment vehicle routing, time windows, local search

JEL classification: C61
AMS classification: 90B06

1 Introduction

Vehicle Routing Problem (VRP) is a combinatorial optimization problem modeling situations in which a nonnegative demand of customers must be satisfied by a fleet of vehicles available at a central depot. It is assumed that each customer can be serviced by exactly one vehicle and the total load of an vehicle cannot exceed its capacity. Each vehicle route must start and end at the depot. The objective is to minimize the total travel cost or alternatively to minimize the number of vehicles used in the solution. The problem is known to be strongly NP-hard. One of many natural extensions of VRP's formulations is the VRP with Time Windows. In this problem, a constraint stating that each customer must be visited within a given time interval must be additionally taken into account. See for example [9] for a survey of VRP and VRPTW variants and solution approaches.

The Multi-Compartment VRP and VRPTW (MCVRP and MCVRPTW respectively) represent generalizations of the classical problem. In MCVRPTW each customer requests a delivery of a set of different products and these products must be transported separately. Vehicles have multiple compartments with limited capacity. Each compartment is dedicated to contain one commodity. The constraint that the demand of each customer must be totally satisfied with a single visit of a vehicle is somewhat relaxed in MCVRPTW. This constraint is reduced only to each commodity while two different commodities are allowed to be delivered in two vehicles. Thus multiple visits of a customer are possible.

Although the problem arises quite often in the practice, it has been seldom studied in the literature. A typical application of MCVRP is the delivery of petroleum products to petrol stations using tank trucks with two compartments (see [2, 3]). In [1] the authors propose a branch-and-price algorithm to solve the MCVRP in which vehicles are equipped with different tanks. Another practical application is the delivery of groceries to convenience stores [4]. Each product requires different temperature, for example low temperature compartment and normal compartment. Animal food distribution to farms is an example of MCVRP mentioned in [5]. The authors propose a memetic algorithm and a tabu search

[1]University of Economics, Prague, Department of Econometrics, nám. W. Churchilla 4, 130 67, Praha 3 , Czech Republic, e-mail: jan.melechovsky@vse.cz

to address the problem. A collection of different kinds of waste is another application studied in [7]. The solution approach is based on a guided local search.

The principal contribution of the paper is the proposition of a metaheuristic algorithm to solve a difficult vehicle routing problem. Moreover, to the best of our knowledge, the problem (MCVRPTW) has not been studied in the literature.

2 Problem formulation

The problem is defined on an undirected graph $G(V, E)$ with $V = \{0, 1, \ldots, n\}$ denoting the set of customer nodes (plus the depot node with index 0) and E representing the set of edges. Each edge $e(i, j)$ has a cost c_{ij} and a travel time t_{ij}. It is assumed that both sets of edge values satisfy the triangle inequality. The depot can supply m different products which can be delivered to customers using a fleet K of identical vehicles with m compartments. Each compartment $p \in \{1, 2, \ldots, m\}$ has a capacity Q_p. Each customer node i has a demand d_{ip} of product p, possibly equal to 0 if product p is not ordered by customer i. A time window $[a_i, b_i]$ is associated with each customer specifying the time slot in which the customer can be visited. The time windows are considered as hard, i.e. earlier arrival than a_i results in a waiting time and later arrival than b_i is not allowed. Moreover, a service time s_i is associated with each visit of customer i. A feasible solution of the problem consists of a set of routes, each starting and ending at the depot, such that the total demand of each product and customer is fully satisfied, the capacity of each vehicle compartment is respected, each delivery to a customer takes place within the given time window and each product is delivered in a single route to each customer. The objective is to find such a feasible solution that the total travel costs are minimized.

3 Evolutionary local search algorithm

As already mentioned the problem is NP-hard and exact solution approaches are rather limited to problems of small size. We are not aware of any application of exact methods to MCVRPTW, but since the problem is harder to solve than VRPTW and the best exact algorithms can deal with instances containing 100 customers and the computing times might grow up to several hours, we can conclude that metaheuristics could be more effective in solving problems of practical size.

The proposed algorithm is an evolutionary local search algorithm (ELS). This method was originally proposed by Merz and Wolf in [6] for a peer-to-peer problem in telecommunications. It extends the classical iterated local search (ILS) which, starting from an initial solution, successively generates child solutions using a perturbation mechanism and local search. The perturbation mechanism serves as a diversification tool while the local search intensifies the search in the current solution neighborhood. ELS additionally generates multiple child solutions and only the best child is kept. The perturbation modifies the solution in a random manner with a parameter r denoting the number of changes made in the solution. If the best child solution improves the best solution found so far the parameter is set to its minimal value r_{min}. If not, r is augmented by a predefined step (e.g. $r = \min(r + 1, r_{max})$).

ELS performs the following steps:

1. Initilize solution x with a simple constructive heuristic.
2. Set $r = r_{min}$ and the best solution $X_{best} = X$.
3. Repeat for a given number of max iterations:
   (a) Initialize best child solution: set $Cost(X'_{best}) = \infty$.
   (b) Repeat for a given number of child solutions:
      i. Initialize child solution: $X' = X_{best}$.
      ii. Perturb X' using perturbation parameter r.
      iii. Apply local search procedure to X'.
      iv. Update best child solution: set $X'_{best} = X'$ if $Cost(X') < Cost(X'_{best})$.
   (c) Update best found solution and perturbation parameter:

- if $Cost(X'_{best}) < Cost(X_{best})$:
  - Set $X_{best} = X'_{best}$.
  - Set $r = r_{min}$.
- else: $r = \min(r+1, r_{max})$

First, an initial solution is determined by a simple heuristic. The perturbation parameter r is initialized to its minimal value and the initial solution enters the ELS loop. ELS performs a given number of iterations with the same value of r. In every iteration, a given number of child solutions is generated using the perturbation mechanism and local search. Only the best child solution is kept in the memory. If the best child solution is better than the best solution found so far, the latter is updated and r is reset to r_{min}. If not, r is incremented by a predefined step.

Initial heuristic

The initial solution is obtained with a simple construction heuristic. The solution is initialized with an empty route. Then all customers with some unsatisfied demand are scanned and the best insertion of a customer into a route is determined. If the customer cannot be inserted into any already existing route, the insertion into a new route is considered instead. The criterion for the insertion is the least increase of travel costs. The procedure terminates when the demand of all customers is fully satisfied.

Perturbation

The perturbation mechanism is one of the two routines of ELS that modify the incumbent solution. It plays the role of a diversification tool in the general ELS framework since it performs several random operations on a solution. Thus it can be interpreted as a mutation operator used in genetic algorithms. The procedure is controlled with a parameter r denoting the number of operations to be performed. In our implementation the operation is a removal and relocation of a customer. For a given route of a solution, r customer nodes are randomly selected and removed from the route. Each of the r removed customers is then tested for a feasible insertion into some of the remaining routes of the solution. If such feasible insertion is detected, the customer is relocated to its new position. If not, new route visiting only this single customer is created.

Local search

Local search is applied to the solution modified by the perturbation mechanism. Its purpose is to improve the solution using a set of operators. It intensifies the search and the improved solution represents a local optimum within the given solution neighborhood. Together with the perturbation mechanism it enables the algorithm to explore effectively the solution space and find potentially good solutions.

Each local search operator defines a solution neighborhood, i.e. the set of solutions that can be obtained by applying the operator. Hence the definition of an operator has a crucial effect to the quality of found solutions. Usually one local search operator is not sufficient to explore the solution space effectively. Combining more operators can increase the probability of finding high quality solutions. On the other hand, the computational complexity shall be also considered.

The implemented local search procedure relies on three operators:

- Path Exchange – exchanges two sub-paths between two routes,
- Relocate – relocates one node within the same route or between two routes,
- Swap – swaps a pair of nodes between two routes.

The operators are applied sequentially with the first improvement strategy – i.e. the first detected improving move is performed. The search stops when no improving move can be found by any operator. Each move must be checked for the feasibility. This involves the time windows as well as the capacity. The time windows feasibility can be checked in $O(1)$ if an information of the maximum feasible shift of each node visit is kept in the memory. However, the update of this information must be done for each node and requires $O(n)$. The capacity must be checked for all vehicle compartments if the move involves two routes. It can be done in $O(1)$ except for Path Exchange where it needs $O(n)$ steps.

4 Preliminary results

The algorithm was coded in C++ and the computational experiments were carried out on a PC equipped with 2.9 GHz dual core processor and 2 GB of RAM. To the best of our knowledge there are no benchmarks proposed in the literature. Therefore the testing environment was derived from existing VRPTW instances.

Test instances

The test instances were derived from the famous VRPTW instances proposed by Solomon in [8]. The data set contains 56 problems divided into six sets: C1, R1, RC1, C2, R2 and RC2. Each instance contains 100 customers plus the depot. The nodes are distributed in a 100 $\times$ 100 square around the depot which is positioned in the middle. Customers are clustered in sets C1 and C2, randomly distributed in sets R1 and R2 and mixed clustered and randomly distributed in sets RC1 and RC2. The position of nodes is identical for all instances within the same set. Only the time window differs. Instances in sets C2, R2 and RC2 are characterized by wider time windows and are reputed as harder to solve.

In [5] the authors have proposed a manner how to derive a MCVRP instances from VRP data sets. We have adapted their idea for the MCVRPTW case. The number of compartments was set to 2. We have considered only sets C1, R1 and RC1 in the preliminary study. The first data set was obtained by splitting the customer demand and the capacity of compartments into two equal parts. The advantage is that any solution feasible for VRPTW remains feasible for MCVRPTW.

The second data set was designed in a way ensuring asymmetric demands of each product. The demand of the first product is calculated as $d_{i1} = d_i/k$, where d_i is the demand of customer i in the original instance and k is a random number from $[3, 5]$. The demand of the second product is $d_{i2} = d_i - d_{i1}$. The values are rounded to nearest integer. The capacity of each compartment is calculated as a function of the total demand and the average demand of each product.

Parameters setting

The ELS algorithm requires only few parameters to be set. The total number of ELS iterations was set proportionally to the number of customers: $N_{max} = 2n$ and the number of child solutions generated within each ELS iteration was set to 10. The perturbation parameter r ranged from 1 to 4.

Results

The computational experiments are still in process so we cannot provide the reader with a detailed analysis of the efficiency of the algorithm. We have performed several tests on the first set of instances, for which the optimal VRPTW solutions are known. The average gap between the ELS solution and the optimum was 4.1 %, 5.8 % and 5.4 % for data sets C1, R1 and RC1 respectively. The average computational times were 79.5 s, 106.7 s and 126.7 s. The average gap is relatively close to the optimum but still it is remarkable that the proposed algorithm is not designed to solve pure VRPTW. We beleive that with further improvements of the local search the solutions might be closer to optima. The average computational times seem to be reasonable when taking into account the difficulty of the problem.

5 Conclusions

The paper introduced a generalization of VRPTW in which multiple vehicle compartments and requested products are taken into account. The problem has not been studied in the available literature. An algorithm based on the evolutionary local search was proposed to address the problem. First results give an optimistic expectations regarding the overall performance of the solution approach. However, further experiments must be carried out in order to obtain better analysis of the performance. Next steps in the research will be focused on the implementation of other operators in the local search and eventually on the design of another metaheuristic framework. A post-optimization phase might be also implemented in order to better explore the promising areas of the solution space.

Acknowledgements

The research was fully supported by the Czech Science Foundation, project reg. number P402/12/P635.

References

[1] Avella, P., Boccia, M., and Sforza, A.: Solving a fuel delivery problem by heuristic and exact approaches, *European Journal of Operational Research* **152** (2004), 170 – 179.

[2] Brown, G. G., Ellis, C. J., Graves, G. W., and Ronen, D.: Real-time, wide area dispatch of mobil tank trucks, *Interfaces* **17** (1987), 107 – 120.

[3] Brown, G. G., and Graves, G. W.: Real-time dispatch of petroleum tank trucks, *Management Science* **27** (1981), 19 – 32.

[4] Chajakis, E. D., and Guignard, M.: Scheduling deliveries in vehicles with multiple compartments, *Journal of Global Optimization* **26** (2003), 43 – 78.

[5] El Fallahi, A., Prins, C., and Wolfler Calvo, R.: A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem, *Computers* & *Operations Research* **35** (2008), 1725 – 1741.

[6] Merz, P., and Wolf, S.: Evolutionary local search for the super-peer selection problem and the p-hub median problem. In: *Lecture notes in computer science* (T. Bartz-Beielstein et al., eds.), volume 4771, Springer Berlin/Heidelberg, 2007, 1 – 15.

[7] Muyldermans, L., and Pang, G.: On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm, *European Journal of Operational Research* **206** (2010), 93 – 103.

[8] Solomon, M. M.: *Vehicle routing and scheduling with time window constraints: models and algorithms.* PhD thesis, University of Pennsylvania, 1983.

[9] Toth, P., and Vigo, D., eds., *The vehicle routing problem.* Society for Industrial and Applied Mathematics, Philadelphia, 2001.