**Lecture Notes in Management Science** (2013) Vol. **5**: 159–166
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

# A variable neighborhood search for the selective multi-compartment vehicle routing problem with time windows

Jan Melechovský

*University of Economics, Prague, Department of Econometrics, Prague, Czech Republic*
*jan.melechovsky@vse.cz*

**Abstract.** This paper presents a generalization of the well-known vehicle routing problem with time windows (VRPTW). In the proposed selective multi-compartment VRPTW (SMCVRPTW) a limited number $k$ of identic vehicles is available at a central depot to serve a set of customers. Each vehicle is equipped with m compartments of limited capacity which are dedicated to transport a particular type of a product. Each customer has a nonnegative demand for up to m products. Once a vehicle delivers a product p to a customer it collects a profit. A vehicle can visit a customer only within a given time window. The SMCVRPTW consist of determining a set of at most k routes starting and ending at the depot, satisfying all customer requests under capacity and time windows constraints such that the total collected profit is maximized. We present a variable neighborhood search algorithm to address the problem. The solution method is evaluated on standard VRPTW benchmarks enhanced with compartments and profit values.

**Keywords:** vehicle routing; time windows; metaheuristic; variable neighborhood search

## Introduction

The selective multi-compartment vehicle routing problem with time windows (SMCVRPTW) generalizes the multi-compartment vehicle routing problem (MCVRP). In the MCVRP, each customer requires the delivery of a nonnegative quantity of product $p \in P$. Products cannot be transported together in one room due to different characteristics. For example refrigerated and freezing compartments are necessary for the delivery of groceries into convenience stores. The customer requests are satisfied either by multiple vehicle visits (each product is delivered separately) or vehicles with multiple compartments can be used instead. The latter situation is modeled by the MCVRP. The problem consists of determining a set of routes visiting customers such that all customer requests are fully satisfied at minimal travel cost under vehicle compartments capacity constraints. Most studies of the MCVRP concern the fuel distribution, e.g. (Brown and Graves, 1981), (Brown et al., 1987), (Avella et al., 2004). Among other practical applications of the MCVRP is the delivery of groceries to convenience stores (Chajakis and Guignard, 2003), distribution of cattle food to farms (Fallahi et al., 2008) or waste collection (Muyldermans and Pang, 2010).

The SMCVRPTW extends the MCVRP. First, each customer can be visited only within a given time window, which is a common additional constraint in various routing problems (e.g. vehicle routing problem with time windows (VRPTW)). Second, a fleet of limited number of vehicles is available at the depot. This constraint implies that some customer requests might remain unsatisfied. Hence the decision problem also involves the selection of a subset of requests that are satisfied. Such a decision criterion can be implemented by assigning a positive profit to each request. Once a request is satisfied, the corresponding profit is collected. The problem can be viewed as a bi-objective optimization problem with one objective maximizing the collected profit and the second objective minimizing the traveled distance. Apart from the pure bi-objective approach, which is out of the scope of this paper, the two objectives are usually handled in three different ways. The first possibility is to maximize the total profit and impose a limit on the maximum route travel cost. The second way is the opposite: minimize the traveled distance under a minimum collected profit constraint. In the third possible formulation, both objectives are present in the objective function. The task is then to maximize the total profit minus the total traveled distance. Each of the formulations constitutes a particular family of routing problems with profits. The corresponding terminology is not unified and similar problems appear under different names in the literature. A classification of traveling salesman problems with profits was provided by (Feillet et al., 2001).

Among recent papers dealing with selective routing problems can be mentioned for example (Valle et al., 2009), (Valle et al., 2011) or (Aras et al., 2011). The first two papers study the Selective Vehicle Routing Problem (SVRP) in which the length of the longest route is to be minimized. The third paper deals with selective multi-depot vehicle routing problem with pricing. The problem models a real-life reverse logistics case. The objective is to maximize the profit minus the cost associated with each collection.

The SMCVRPTW presented in this paper is motivated by the following real-life application. Consider the case of the delivery of groceries to convenience stores. The deliveries must be quite frequent since the available stock is rather low. Simultaneously, each supply meets several limitations. The unloading is usually done on the pavement in front of the store and it affects the store's operation. As well parking capacities in urban areas are restricted and a situation when multiple vehicles arrive at the same time can be troublesome. Vehicles equipped with multiple compartments suitable to transport all required kinds of goods can reduce the number of visits to a customer. Furthermore, imagine the case when only a limited number of such vehicles is available. The interest of the distributor would be then to maximize the number of satisfied requests at minimum travel cost. This is the case modeled by the proposed SMCVRPTW.

The problem is NP-hard since it generalizes the vehicle routing problem. Exact algorithms can be used only to problems of a moderate size. Hence this paper focuses on a metaheuristic approach to address the problem.

The paper presents a first study on the SMCVRPTW. The next section formulates the problem. Then a metaheuristic approach based on the variable neighborhood search is presented. Preliminary results and conclusion encloses the paper.

## Problem formulation

The SMCVRPTW is defined on a complete undirected graph *G(V,E)*, in which $V = \{0, 1, \ldots, n\}$ represents the set of nodes and $E$ the set of edges. The node 0 represents the depot. Each customer $i \in V \setminus \{0\}$ requires the delivery of nonnegative quantity $d_{ip}$ of product $p \in P$. We call the pair $(i, p)$ the request in the sequel. The delivery must take place only within a given time window $[e_i, l_i]$. Earlier arrival results into a waiting time of the vehicle while later arrival is forbidden. Each required product must be delivered on a single vehicle to the customer; however different products for the same customer can be loaded into different routes. Once product $p$ is delivered to customer $i$, profit $r_{ip} \geq 0$ is collected. We define $R_i = \sum_{p \in P} r_{ip}$ the total profit of customer $i$. Given a solution $s$ we also define $R(s)$ as the total profit collected in $s$. A fleet of $m$ identic vehicles is available at the depot. Vehicles are equipped with $w = |P|$ compartments with limited capacity $Q_p$. Nonnegative travel cost $c_{ij}$ and nonnegative travel time $t_{ij}$ is associated with each edge $(i, j) \in E$ . It is assumed that both sets of values satisfy the triangle inequality. The aim is to determine a set of at most $m$ vehicle routes serving a subset of customer requests such that the total profit is maximized and the total traveled distance (with respect to $R(s)$) is minimized. Given two solutions $s_1$ and $s_2$, we say that $s_1$ dominates $s_2$ ($s_1 \prec s_2$) if $R(s_1) > R(s_2)$ or $R(s_1) = R(s_2)$ and the total travel cost $c(s_1) < c(s_2)$.

## Initial solution heuristic

A simple insertion heuristic is used to initialize the solution. In the first step, $m$ routes are built with the $m$ most profitable customers (according to $R_i$). If $R_i$ is identic for more customers, the travel cost is considered as the decision criterion guiding the selection. The solution is completed with further customer requests as follows. For each customer node $i$, let $A_i$ denote the set of customer nodes reachable from $i$ in terms of time windows. Let $g_i = |A_i|$ be the number of elements in $A_i$. Furthermore, let $N$ denote the subset of customers that are not inserted in any route yet. In an iteration, each feasible insertion of customer $j \in N$ in a route between nodes $i \notin N$ and $k \notin N$ is evaluated according to the ratio $\rho_{ij} = g_j \frac{R_j}{\Delta_{ij}}$. The term $\Delta_{ij} = t_{ij} + t_{jk} + w_j - t_{ik}$ measures the possible time spent when visiting customer $j$. Note that $w_j$ is the waiting time at $j$ when arriving from $i$. The customer with maximum ratio is selected and the insertion is performed. In this step, the insertion involves all products demanded by the inserted customer and thus the total profit of the customer is collected. The procedure is repeated until no feasible insertion can be found further. The insertion selection based on the ratio $\rho_{ij}$ aims to find a reasonable tradeoff between the total customer's profit $R_i$ and its connectivity. The term $g_i$ encourages customers having more reachable neighbors, while $\Delta_{ij}$ tries to favor customers whose insertion consumes less time. The insertion heuristic achieved better results on average when the ratio $\rho_{ij}$ was used compared to a simple maximization of the inserted profit.

```
Input:      An instance of the SMCVRPTW.
Output:     A feasible solution x* maximizing
            the total collected profit.
  x* ← InitialHeuristic()
  k ← 1
  i ← 0
  ni ← 0
  while (i < i_max) and (ni < ni_max) do:
      x′ ← RandomRemove(k)
      x′′ ← LocalSearch()
      if (x′′ ≺ x*):
            x* ← x′′
            k ← 1
            ni ← 0
      else:
            k ← min(k+1,k_max)
            ni ← ni + 1
      i ← i+1
```

## Variable neighborhood search solution approach

Variable neighborhood search (VNS) is a metaheuristic framework proposed by (Mladenovic and Hansen, 1997). Its principle idea is to dynamically change neighborhood when a local optimum is reached. The VNS consists of a local search mechanism, a set of different neighborhoods and a set of rules defining the use and the selection of neighborhoods according to search results. In the present implementation of the VNS, the set of neighborhoods $N_k(x), k = 1, \ldots, k_{max}$ of a solution $x$ is defined as the set of all solutions that can be obtained by removing a sequence of $k$ consecutive nodes from $x$. The search starts with $k = 1$. Given an incumbent solution $x^*$, $k$ consecutive nodes are removed at random from each route of $x^*$ . Let $x'$ be the solution obtained from $x^*$. The local search, applied to $x'$, tries to refill the routes in $x'$ with yet unsatisfied requests maximizing the collected profit. If the resulting solution $x''$ dominates $x^*$, the search continues with $x''$ as the incumbent solution ($x^* = x''$) and $k = 1$. If not, the neighborhood is enlarged by setting $k = \min(k + 1, k_{max})$ and the search restarts with $x^*$unchanged. The steps of the VNS are depicted in Algorithm 1. The algorithm stops when either the maximum number *i_max* of iterations is encountered or *ni_max* iterations without improvement of the incumbent solution are performed.

The main focus of the local search mechanism is to find a solution with the maximum total profit in the current neighborhood. A secondary objective is to minimize the traveled distance. The implemented local search thus disposes of two subsets of operators, each enabling to handle one objective. The first subset consists of common routing moves:

*a) 2-opt – replaces two edges in a single tour and inverts a subsequence of nodes,*
*b) 2-opt* – swaps two sub-paths between two routes,*
*c) Relocate – relocates a subsequence of nodes,*
*d) Swap – swaps two subsequences of nodes.*

Note that *2-opt* operates only on a single route while *2-opt** makes the exchange of two sub-paths between two distinct routes. *Relocate* and *Swap* can operate either on a single route or on a pair of routes. Inter-route operations b) – d) are evaluated in this order and the first improvement is performed. When a route is modified, it is checked for further improvements with *2-opt* and the single route *Relocate* and *Swap*. Moreover, these two operators can explore different neighborhoods defined by the number $q$ of consecutive nodes that are displaced at one time. Both implementations of *Relocate* and *Swap* are initialized with $q = 1$. If an improving move is determined, the current solution is modified and the search continues by exploring another routing move in the sequel. If not, *Relocate* or *Swap* is further explored with $q = \min(q + 1, q_{max})$.

The second subset of operators enables the insertion of requests which are not satisfied in the current solution. Given a route $s$ and a parameter $q'$, the operator consists of a removal of $q'$ consecutive nodes from $s$ and its replacement with a new sequence of requests. Note that when a node is removed from a route then the entire quantity of each delivered product is removed altogether. Contrarily, the entering sequence considers individual requests: if a customer is visited, only a subset of the demanded products can be delivered. This operator hence allows for a partial splitting of deliveries. The improvement procedure is initialized with $q' = 0$ (i.e. no customer is removed). Each route of the current solution is scanned and possible replacements of satisfied customers/requests are examined. A move is accepted only if the difference between the inserted profit and the removed profit is strictly positive. The first acceptable move is performed and the local search procedure starts again with the sequence of b) – d) operators. If no improving move can be determined with the current value of $q'$, the search continues in inspecting possible replacements with $q' = \min(q' + 1, q'_{max})$.

The determination of the entering sequence can be modeled as the elementary shortest path problem with resource constraints (ESPPRC) as follows. Let $G'(V', E')$ be a complete undirected graph with $V' = N \times P$ representing requests. An edge $(u, v) \in E'$ has a cost $\lambda_{uv} = -r_{(i,p)\in v}$, a travel time $\tau_{uv} = t_{ij:(i,\cdot)\in u,(j,\cdot)\in v}$ and a travel distance $\gamma_{uv} = c_{ij:(i,\cdot)\in u,(j,\cdot)\in v}$. In words, the cost is defined as the negative profit collected when going from $u$ to $v$, while the travel time and the travel distance correspond both to the appropriate values in the original graph $G$. We also assume that $t_{ii} = 0$ and $c_{ii} = 0$ for each $i \in G$. When an edge $(u, v)$ is used in the path, resources are consumed. The resources that must be taken into account are the available time slot in the examined route and the capacities of the vehicle compartments. Optionally the original travel cost might be also considered. The resulting ESPPRC then consists of finding an elementary path (in the sense that no request can be visited more than once), respecting the resource limits and maximizing the total collected profit. The problem is known to be NP-hard. However, Feillet et al., 2004 proposed an effective label correcting algorithm for the ESPPRC. A similar implementation was used in the presented local search procedure. Although this approach is time consuming in general, the restricted solution space implied by the fact only a fraction of route is considered in every ESPPRC calculation renders the computation tractable.

## Preliminary results

The first computational analysis of the proposed VNS was carried out on a set of test problems derived from the well-known Solomon's benchmarks originally proposed for the VRPTW (Solomon, 1983). 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 x 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 travel cost and the travel time is calculated as the Euclidean distance, the latter rounded to one decimal. Each customer $i$ has associated a randomly distributed demand $d_i$ and a time window $[e_i, l_i]$. Instances in sets C2, R2 and RC2 are characterized by wider time windows and are reputed to be harder to solve.

The SMCVRPTW test instances were generated as follows. The number of compartments $w$ was set to 2. The problem is already difficult with $w = 1$ and testing with larger values will be the task of further research. The demand of customer $i$ for product 1 was calculated as $d_{i1} = \frac{d_i}{\mu}$, where $\mu \in [3,5]$ is a random value. The demand for the second product was calculated as $d_{i2} = d_i - d_{i1}$. The capacity of compartments was calculated as $Q_p = (Q \times D_p)/(D_1 + D_2)$, where $Q$ is the vehicle capacity in the original VRPTW and $D_p$ is the average demand for product $p$. The profit $r_{ip}$ was set to 1 equally for all requests. Finally, the number m of available vehicle ranged from 1 to 4.

The stopping condition of the algorithm was met when 1 500 iterations were encountered or if the number of non-improving iterations was equal to 300. The parameters used in the definition of solution neighborhoods were set as follows: $k_{max} = 7$, $q_{max} = 3$ and $q'_{max} = 2$.

| Data set | # problems | m = 1 | | | m = 2 | | | m = 3 | | | m = 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Profit | Travel distance | Time (s) | Profit | Travel distance | Time (s) | Profit | Travel distance | Time (s) | Profit | Travel distance | Time (s) |
| C1 | 9 | 18 | 102.3 | 85.3 | 32 | 174.3 | 120.5 | 45 | 296.2 | 188.2 | 87 | 320.4 | 205.6 |
| C2 | 8 | 21 | 204.9 | 96.6 | 39 | 326.5 | 147.2 | 59 | 458.3 | 174.2 | 96 | 550.6 | 226.7 |
| R1 | 12 | 11 | 196.2 | 75.4 | 28 | 386.4 | 153.9 | 42 | 496.8 | 193.1 | 75 | 578.1 | 209.3 |
| R2 | 11 | 19 | 130.8 | 86.7 | 27 | 247.2 | 171.2 | 38 | 396.7 | 198.6 | 59 | 472.3 | 211.8 |
| RC1 | 8 | 10 | 156.7 | 68.1 | 18 | 312.6 | 136.8 | 36 | 465.1 | 148.1 | 61 | 512.6 | 201.2 |
| RC2 | 8 | 15 | 178.2 | 74.2 | 22 | 296.1 | 145.2 | 39 | 387.3 | 159.3 | 58 | 503.7 | 196.7 |

**Fig. 1.** Average results on SMCVRPTW instances.

The computational experiments were carried out on a notebook equipped with a 2.5 GHz dual core processor. The results were obtained with 5 runs, each time with a different rand seed setting. Table 1 presents the average results obtained per each data set. The first column contains the name of the instance set. In the second column is reported the number of problems solved within an instance set. The results are then reported for each $m$. The Table shows the first results obtained with the proposed VNS and therefore it is difficult to provide a detailed analysis at this moment, since there are no solutions to compare with. However, the average computing times at least show that the algorithm is capable to solve the problem within a reasonable time. The profit and the distance increases with increasing $m$, which is of course an expected behavior. Further experiments will be carried out in order to present a more complex analysis.

## Conclusion

This paper presented the first of the SMCVRPTW. The problem involves several decision problems which makes the SMCVRPTW difficult to solve. However, it is relevant practical problem often encountered in a daily distribution of different kinds of goods. A metaheuristic solution based on the VNS was developed to address the problem. The key idea of the implemented algorithm is to explore new solution regions performing exchanges of satisfied and unsatisfied requests in the incumbent solution. A key component of the VNS is the local search which enables to optimize two objectives (the profit and secondary the travel cost) simultaneously and which uses an exact solution approach to determine the locally optimal sequence of requests.

Preliminary results were reported, however, more complex experiments must be carried out. The experiments will involve more test problems with larger values for the number of products and non-trivial values of the collected profits. Further research will be focused especially on the evaluation of ESPPRC within the local search framework and the benefits (losses) of the exact approach compared to some simple heuristic mechanism. A comparison with other solution approaches is also on the agenda.

**Acknowledgments**—The research was fully supported by the Czech Science Foundation under research projects P402/12/P635 and P403/12/1947. This financial support is kindly acknowledged.

## References

Aras N, Aksen D and Tekin MT (2011). Selective multi-depot vehicle routing problem with pricing. Transportation Research Part C: Emerging Technologies, 19(5):866 – 884.
Avella P, Boccia M and Sforza, A (2004). Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research, 152(1):170 – 179.
Brown G G, Ellis C J, Graves GW and Ronen D (1987). Real-time, wide area dispatch of mobil tank trucks. Interfaces, 17(1):107 – 120.
Brown GG and Graves GW (1981). Real-time dispatch of petroleum tank trucks. Management Science, 27(1):19 – 32.
Chajakis ED and Guignard M (2003). Scheduling deliveries in vehicles with multiple compartments. Journal of Global Optimization, 26:43 – 78.
Fallahi AE, Prins C and Calvo RW (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers & Operations Research, 35(5): 1725 – 1741.
Feillet D, Dejax P and Gendreau M (2001). Traveling salesman problems with profits: An overview. Transportation Science, 39:188–205.
Feillet D, Dejax P, Gendreau M and Gueguen C (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks, 44(3): 216–229.
Mladenovic N and Hansen P (1997). Variable neighborhood search. Computers & Operations Research, 24(11):1097 – 1100.
Muyldermans L and Pang G (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. European Journal of Operational Research, 206(1):93 – 103.
Solomon M M (1983). Vehicle routing and scheduling with time window constraints: models and algorithms. PhD thesis, University of Pennsylvania.
Valle CA, da Cunha AS, Mateus GR and Martinez LC (2009). Exact algorithms for a selective vehicle routing problem where the longest route is minimized. Electronic Notes in Discrete Mathematics, 35(0):133 – 138.
Valle CA, Martinez LC, da Cunha AS and Mateus GR (2011). Heuristic and exact algorithms for a min - max selective vehicle routing problem. Computers & Operations Research, 38(7):1054 – 1065.