A Multi-restart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle Routing Problem with Time Windows

Olli Bräysy
Agora Innoroad Laboratory, P.O.Box 35, FI-40014 University of Jyväskylä, Finland
Wout Dullaert
Institute of Transport and Maritime Management Antwerp, University of Antwerp, Keizerstraat 64, B-2000 Antwerp, Belgium
Geir Hasle
SINTEF ICT, P.O. Box 124 Blindern, NO-0314 Oslo Norway
David Mester
Institute of Evolution, Mathematical and Population Genetics Laboratory, University of Haifa, 31905 Haifa, Israel
Michel Gendreau
Centre de recherche sur les transports, Université de Montréal, Case postale 6128, Succursale ”Centre-ville”, Montréal, Canada H3C 3J7

Abstract

We present a new deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. The objective is to service a set of customers at minimal cost within their time windows by a heterogeneous capacitated vehicle fleet. The suggested metaheuristic comprises three phases. In the first phase, high quality initial solutions are generated by means of a savings-based heuristic combining diversification strategies with learning mechanisms. An attempt is then made to reduce the number of routes in the initial solution with a new local search procedure in phase two. The solution is further improved in phase three by a set of four local search operators that are embedded in a deterministic annealing framework. Computational experiments show that the suggested method outperforms the previously published results.

1 Introduction

The Vehicle Routing Problem (VRP) is a key to efficient transportation management and supply-chain coordination. Although often assumed in VRP theory, a vehicle fleet is rarely homogeneous in real life. A fleet manager typically controls vehicles that differ in their equipment, carrying capacity, speed, and cost structure. There are many types of motivation for keeping a diversified fleet. Some customers may require vehicles with costly equipment. Large vehicles may be more cost effective, but the road network and the nature of a customer’s premises often impose physical restrictions on vehicle size and weight. Some types of vehicle may not be allowed in urban areas due to environmental concerns. Vehicles of different carrying capacity give flexibility to allocate capacity according to varying demand in a more cost effective way.

The Fleet Size and Mix Vehicle Routing Problem (FSMVRP) is a VRP where the homogeneous fleet assumption of the traditional VRP has been lifted. In this paper, we study the FSMVRP with time windows (FSMVRPTW), where each customer has a time window within which service must start. It is a natural extension of the recently much studied VRPTW (see Bräysy and Gendreau, 2005a and 2005b). Time windows are critical in many applications, as more and more importance is given to customer service and timeliness. We focus on the FSMVRPTW variant that was first studied by Liu and Shen (1999). Here, the fleet is heterogeneous because each vehicle might have its own fixed cost and its own capacity. Driving speed and customer service times are identical for all vehicles. The routing cost for a solution is defined as the sum of "en route time" over all vehicles used. The main contribution of this paper is to present a new efficient multi-start deterministic annealing (MSDA) heuristic for the problem. In Section 1.1, a mathematical formulation of the FSMVRPTW is given and Section 1.2 presents a literature review. Section 2 contains a description of the algorithmic approach. Section 3 describes our experimental study and results. Conclusions are given in Section 4.

1.1. Problem Formulation

We base our formulation of the FSMVRPTW on the well-known vehicle flow arc-based formulation for the VRPTW. Let $\mathbf{G} = (\mathbf{N}, \mathbf{A})$ be a graph, where $\mathbf{N} = \{0\} \cup \{1, \ldots, n\} \cup \{n+1\}$. 0 and $n+1$ represent the depot. $\mathbf{C} = \{1, \ldots, n\}$ is the set of customers. $\mathbf{A} \subseteq \mathbf{N} \times \mathbf{N}$ represents the travel possibilities between nodes. $\mathbf{V} = \{1, \ldots, K\}$ is the set of alternative vehicles. There are vehicle specific acquisition / depreciation costs e^k and capacities q^k. t_{ij} are the vehicle independent travel time between nodes. For each customer i there is a time window $\left[a_i, b_i\right]$, and s_i is the vehicle independent customer service time. For each (i, j, k) with $i, j \in \mathbf{A},\ k \in \mathbf{V}, x_{ij}^k$ is a binary decision variable that expresses whether vehicle k travels directly from customer i to customer j. For each (i,k), $i \in \mathbf{A},\ k \in \mathbf{V},\ y_i^k$ determines the exact start time of service at this customer if it is served by that vehicle.

$$\text{minimize} \quad \sum_{k\in\mathbf{V}}\sum_{j\in\mathbf{C}}\left(e^k + y^k_{n+1} - y^k_0\right)x^k_{0j} \qquad (1)$$

subject to:

$$\sum_{k\in\mathbf{V}}\sum_{j\in\mathbf{N}} x^k_{ij} = 1, \qquad \forall i \in \mathbf{C} \qquad (2)$$

$$\sum_{i\in\mathbf{C}} d_i \sum_{j\in\mathbf{N}} x^k_{ij} \le q^k, \qquad \forall k \in \mathbf{V} \qquad (3)$$

$$\sum_{j\in\mathbf{N}} x^k_{0j} = 1, \qquad \forall k \in \mathbf{V} \qquad (4)$$

$$\sum_{i\in\mathbf{N}} x^k_{ih} - \sum_{j\in\mathbf{N}} x^k_{hj} = 0, \qquad \forall h \in \mathbf{C}, \ \forall k \in \mathbf{V} \qquad (5)$$

$$\sum_{i\in\mathbf{N}} x^k_{i,n+1} = 1, \qquad \forall k \in \mathbf{V} \qquad (6)$$

$$x^k_{ij}(y^k_i + s_i + t_{ij} - y^k_j) \le 0, \qquad \forall (i,j) \in \mathbf{A}, \ \forall k \in \mathbf{V} \qquad (7)$$

$$x^k_{ij}\left(y^k_i - a_i\right) \ge 0, \qquad \forall (i,j) \in \mathbf{A}, \forall k \in \mathbf{V} \qquad (8a)$$

$$x^k_{ij}\left(b_i - y^k_i\right) \ge 0, \qquad \forall (i,j) \in \mathbf{A}, \forall k \in \mathbf{V} \qquad (8b)$$

$$x^k_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathbf{A}, \ \forall k \in \mathbf{V} \qquad (9)$$

Constraints (2) state that all customers must be visited by a vehicle. (3) are the vehicle capacity constraints. Constraints (4) enforce that each vehicle must leave the depot exactly once, and (6) that each vehicle must arrive at the depot exactly once. (5) ensure that if a vehicle arrives at a customer, it also departs from that customer. (7) guarantee that the arrival times at two consecutive customers allow for service and travel time. (8a) and (8b) are the time window constraints. The non-linear travel time and time window constraints can be easily linearized by use of the “big M” trick. The FSMVRPTW is strongly **NP**-hard as it is a generalization of the CVRP. In this paper, we focus on heuristics.

1.2. Literature review

The first FSMVRPTW work was reported by Liu and Shen (1999). They developed a savings-based construction heuristic and an improvement heuristic inspired by the work of Golden et al. (1984) for the FSMVRP. The approach was tested on a new benchmark developed by the authors based on a well-known benchmark for the VRPTW (Solomon, 1987). Three sets of instances were created, each representing a different cost structure for vehicle types. Dullaert et al. (2002) proposed a sequential insertion heuristic based on Solomon’s I1 heuristic for the VRPTW and tested their approach on the Liu and Shen benchmark. More recently, the FSMVRPTW has been studied by Privé et al. (2006), Dell’Amico et al. (2006), Calvete et al. (2006), Tavakkoli-Moghaddam et al. (2006), and Dondo and Cerdá (2006).

2 Solution approach

Our proposed solution approach consists of three phases. In the Phase 1 high quality initial solutions are generated by means of a savings-based heuristic combining diversification strategies with learning mechanisms. In Phase 2 the focus is on reducing the number of vehicles. A set of four local search operators are embedded in a deterministic annealing framework to guide further improvement in Phase 3. All phases are repeated a user-defined number of times, i.e., several initial solutions are created and improved separately.

2.1. Phase 1: Constructing initial solutions

Initial solutions are created by a probabilistic modification of the savings heuristic (Clarke and Wright, 1964). The search is started by serving each customer by way of a separate route. After the first construction and improvement phase, some of the arcs in the final solution of the previous improvement phase are kept instead of starting with single customer routes in Phase 1.

The route construction heuristic features three main modifications compared to the original savings heuristic. First, the heuristic is implemented from an insertion point of view to combine the strengths of savings and insertion-based heuristics. When merging two routes R_1 and R_2, all positions between consecutive customers within route R_2 are considered. Second, the algorithm accepts not only the best savings, but also the second and third best savings (if positive) with given probabilities, to allow for diversification. Third, each route is initialized with the smallest possible vehicle type. Savings are calculated by considering both fixed vehicle costs and the total schedule time. Vehicle sizes are updated whenever needed, and are always set to the smallest vehicle available capable of serving the customers on the route. Merges are attempted until no further improvement can be found.

2.2. Phase 2: Route Elimination

Route elimination is based on a depletion procedure called ELIM, which again uses simple insertions. All routes of the current solution are considered for depletion, in random order. For a given route, ELIM removes all customers and tries to insert them in the remaining routes. The criticality measure ς_i defined below determines the sequence in which customers are reinserted. The bars mean normalization to $[0,1]$.

$$\varsigma_i = \frac{\overline{d_i}}{\overline{(b_i - a_i)}} + \overline{c_{oi}}$$

For a given customer, all positions in the remaining routes are tried. The best feasible insertion according to the total cost objective is selected. The algorithm starts by calculating total savings obtained by eliminating the currently selected route. It then keeps track of the total increase in total costs resulting from reinsertions. If the total increase exceeds the total saving or if any of the customers cannot be inserted in another route, the algorithm reverts and proceeds to the next route in sequence. If all the removed customers have been inserted in other routes at a lower cost, the route is eliminated. In the Route Elimination Phase, ELIM is run until quiescence.

2.3. Phase 3: Local Search Improvement

The solution from Phase 2 is further improved by local search and metaheuristics. Four local search operators are utilized: ELIM described above, a route splitting operator called SPLIT, along with variants of the ICROSS and IOPT operators suggested in Bräysy (2003). The SPLIT neighborhood consists of all solutions that result from splitting a single route in the current solution into two parts at any point. We employ it in a greedy, first-accept fashion, simply by looping through all routes and all customers in them, splitting the selected route into two parts at the position of the current customer. The move is made if the split reduces cost. ICROSS has been extended with the adjustment of vehicle types.

The four local search operators are employed as follows: ICROSS, IOPT, ELIM, and SPLIT (in that order) are repeated for a given number of iterations $n_{improve}$. Before each iteration, the routes are randomly ordered. The search is embedded in a Threshold Accepting (TA) metaheuristic (Dueck and Scheurer, 1990). Our procedure starts with threshold $t = 0$ (no worsening) and is repeated with that value until a local minimum is reached. If no improvement has been found for a given number of iterations $\bar{n}$, t is set to a maximum value and the search is restarted from the current best solution. The maximum value is $r \cdot t_{\max}$ where r is a random number in the range [0,1] and $t_{\max}$ is a user-defined parameter. At each non-improving iteration, t is reduced by Δt until zero is reached. The search is repeated with zero threshold until no more improvements can be found. To escape the local optimum, t is set to a new maximum threshold value $r \cdot t_{\max}$. To limit search effort for ICROSS and IOPT, we focus only on moves that involve customers that are close. When creating subsequent initial solutions, the algorithm uses information gathered during the search in a frequency-based long-term memory of arcs that appear in high quality solutions. This learning mechanism is based on identifying the common arcs in the sequence of incumbent solutions.

When the iteration limit $n_{improve}$ for the current initial solution is reached, a starting point is created for the savings heuristic to create a new initial solution based on arc frequency information. If an arc is not present in a user-defined share of the best solutions, it is removed. Remaining arcs are randomly deleted or kept according to a given user-defined probability. The resulting arc set forms a new starting point for the savings algorithm. The creation of initial solutions and subsequent improvement is repeated for a given number of times, n_{init}, specified by the user.

3 Computational experiments

The proposed algorithm was implemented in Java (JDK 5.0) and tested on an AMD Athlon 2600+ (512 MB RAM) computer. Experiments were performed on the instances proposed by Liu and Shen (1999), who introduced several vehicle types with different capacities and costs to the 56 Solomon 100 cases. In addition, three different vehicle cost structures *A*, *B* and *C* were suggested, resulting in 168 problem instances.

After sensitivity analysis the following parameter values were used: $n_{improve} = 1000$; $n_{init} = 2$; $t_{\max} = 0.04$; $\Delta t = 0.001$; $\overline{n} = 40$. It appeared to be important to create several initial solutions, but the gain from additional solutions becomes small after 5-6 solutions. Three runs were used for each instance. Results show that the MSDA procedure is consistently better than the competition for all data sets. It found 157 best-known solutions, where 6 are ties and 151 are new solutions. Considering computer speed and the number of runs, computing times are slightly higher than the competition. A quick variant with a single run and 200 iterations is significantly faster and gives better average quality than the competition for all data sets.

4 Summary and Conclusions

The Fleet Size and Mix Vehicle Routing Problem with Time Windows (FSMVRPTW) is an industrially important problem that has not been studied much in the literature. We have developed a novel 3-phase algorithm for solving the FSMVRPTW. With a moderate increase of computational effort relative to the competition, our procedure significantly outperforms the competition with a small increase of effort and has found 151 new best known solutions.

References

Bräysy, O. 2003. A reactive variable neighborhood search for the vehicle routing problem with time windows. INFORMS J. Comput. 15 347–368.

Bräysy O., M. Gendreau. 2005a. Vehicle routing problem with time windows, Part I: route construction and local search algorithms. Transportation Sci. 39 104–118.

Bräysy O., M. Gendreau. 2005b. Vehicle routing problem with time windows, Part II: metaheuristics. Transportation Sci. 39 119–139.

Calvete, H.I, C. Galé, M.-J. Oliveros, B. Sánchez-Valverde. 2006. A goal programming approach to vehicle routing problems with soft time windows. To appear Eur. J. of Oper. Res.

Clarke G., J.W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. 12 568–581.

Dell'Amico M., M. Monaci, C. Pagani, D. Vigo. 2006. Heuristic approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. Working paper, University of Modena, Italy.

Dondo R, J. Cerdá. 2006. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. To appear Eur. J. Oper. Res.

Dueck G, T. Scheurer. 1990. Threshold accepting: a general purpose optimization algorithm. J. Comput. Phys. 90 161–175.

Dullaert W, G.K. Janssens, K. Sörensen, B. Vernimmen. 2002. New heuristics for the fleet size and mix vehicle routing problem with time windows. J. Oper. Res. Soc. 53 232–1238.

Golden B, A. Assad, L. Levy, F. Gheysens. 1984. The fleet size and mix vehicle routing problem. Comput. Oper. Res. 11 49–66.

Liu F-H, S.-Y. Shen. 1999. The fleet size and mix vehicle routing problem with time windows. J. Oper. Res. Soc. 50 721–732.

Privé J, J. Renaud, F.F. Boctor, G. Laporte. 2006. Solving a vehicle routing problem arising in soft drink distribution. J. Oper. Res. Soc. 57 1046–1052.

Solomon MM. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 254–265.

Tavakkoli-Moghaddam R, N. Safaei, Y. Gholipour. 2006. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. Appl. Math. Comput. 176 445–454.