Fast Approximation Heuristics for Multi-Objective Vehicle Routing Problems

Martin Josef Geiger

Helmut Schmidt University, University of the Federal Armed Forces Hamburg,
Logistics Management Department, Holstenhofweg 85, 22043 Hamburg, Germany
m.j.geiger@hsu.hh.de
http://logistik.hsu-hh.de/geiger

Abstract. The article describes an investigation of the use of fast approximation heuristics for multi-objective vehicle routing problems (MO-VRP). We first present a constructive heuristic based on the savings approach, which we generalize to fit the particular multi-objective nature of the problem. Then, an iterative phase based on local search improves the solutions towards the Pareto-front. Experimental investigations on benchmark instances taken from the literature show that the required computational effort for approximating such problems heavily depends on the underlying structures of the data sets. The insights gained in our study are particularly valuable when giving recommendations on how to solve a particular MO-VRP or even a particular MO-VRP instance, e. g. by means of a posteriori or interactive optimization approaches.

1 Introduction

Since the early development of operations research techniques for problems found in logistics, a considerable progress can be observed with solving challenging optimization models from that domain. Nowadays, (semi-)automated planning systems are available for a number of logistical applications. A prominent example can be found in vehicle routing, where a given set of transportation orders has to be served using a fleet of vehicles. The solution to such problems generally involves the assignment of orders to resources (vehicles), as well as the routing of the vehicles, and a common optimality criterion is found in the minimization of the total cost of transportation. Besides, complex side constraints have to be obeyed, such as time windows defining the availability of resources or expressing preferences of the customers. Often, such problems are referred to as 'rich' vehicle routing problems.

Practical vehicle routing problems not only involve numerous constraints, but also other criteria besides the cost minimization objective, e. g. the maximization of the provided service. Consequently, multi-objective problem formulations are proposed in the literature. The simultaneous consideration of conflicting objectives has some effects on the solution of vehicle routing problem formulations. Since not a single optimal solution exists that optimizes all criteria at once, a Pareto-set P containing equally Pareto-optimal alternatives must be found, and

C. Di Chio et al. (Eds.): EvoApplications 2010, Part II, LNCS 6025, pp. 441–450, 2010.

the choice of a most-preferred solution $x^* \in P$ must be made involving a human decision-maker.

Solution strategies for multi-objective vehicle routing problems can be organized in three ways.

1. *A priori* approaches first reduce the multi-objective formulation into a single-objective one by introducing a utility function or some other approach with which the preference of the decision maker can be expressed and measured.
2. *A posteriori* approaches on the other hand identify the Pareto-set P in an offline optimization phase, and then allow the decision maker to identify some most-preferred element $x^* \in P$.
3. *Interactive* approaches aim to combine the above mentioned concepts by allowing a gradual articulation of preferences, alternating between optimization and decision making phases until some satisfying alternative is found.

Obviously, both the optimization and the decision making phase are time-consuming processes, and time is, especially in operative planning problems, a limited resource. Therefore, any practical implementation of a solution concept must find the correct balance of each part. From that perspective, no general recommendation of which of the three concepts should be used can be given.

(Meta-)heuristics have become increasingly popular for solving vehicle routing problems [1,2,3,4]. More recently, multi-objective adaptations of such techniques have been proposed and applied to the VRP [5, 6, 7]. Interactive approaches are however still rather new and need to be studied further [8].

In the light of the explanations above, it becomes clear that more thorough investigations are needed with respect to two issues. (i) First, fast approximation approaches are needed, that allow the (approximation) solution of hard combinatorial optimization problems within little time. Such ideas generally contribute to the development of interactive approaches, that have to rely on decision support systems providing fast responses. (ii) Second, the cost of approximating multi-objective vehicle routing problems using heuristic search algorithms must be further studied, with 'cost' measuring the required computations effort for finding (local) optima. The results obtained from the investigation may then be used when trying to give recommendations on whether to opt for a priori, a posteriori, or interactive approaches in the particular problem domain. As we cannot expect to observe an algorithmic behavior independent from the underlying characteristics of the particular benchmark instances, structurally different data sets are used for the experiments.

2 Problem Statement

The vehicle routing problem under multiple objectives as tackled in this article is defined by a given set of customers, each of which must be served with a given number of goods from a depot by means of a vehicle. It therefore falls into the class of deterministic problems, an assumption which appears on a short planning horizon to be feasible in many practical cases. A formal problem representation is possible using a graph $G = (V, E)$, consisting of vertices $v_i \in V$ representing customers i at given locations, and an interconnecting edge set E representing the route network between the nodes, and traveling along the edges of the graph results in a given travel time. For each customer i, time windows $[t_i^e, t_i^l]$ are given, which define the desired time of service. With respect to our multi-objective formulation of the problem, the earliest possible time of service t_i^e is interpreted as a constraint. The lasted desired time t_i^l however may be violated, but is penalized by means of an additional objective function as described below [9].

Additional side constraints of the problem are capacity constraints of the vehicles and their maximum total travel time $t^{\max}$, which easily can be represented by a (hard) time window at the depot node v_0 by $[t_0^e = 0, t_0^l = t^{\max}]$. We further require that all customers are serviced by exactly one vehicle, therefore avoiding split-deliveries.

A feasible solution to the problem defines routes and thus delivery schedules for each customer, respecting all given side constraints of the problem. Alternatives are evaluated with respect to two criteria: The minimization of the resulting cost and the maximization of the quality of the provided of service. While the first criterion is translated into the minimization of the total traveled distances, the second is represented by the total tardiness, thus providing a measure for the punctuality of the deliveries.

Besides the scope of this investigation, other criteria might be of relevance also, e. g. the minimization of the number of vehicles in use. On a short planning horizon however, the available fleet for transportation is often constant, and therefore is left aside here.

A considerable conflict among the two objective functions can be expected. A simple argument for this is that a direct delivery to each customer will yield into a minimum achievable tardiness, but will inflict high cost. Any practical solution consequently must find a balance between the two criteria, and a human decision maker will have to state his/her preferences with respect to the two issues in order to identify a preferred compromise solution. From a decision support and optimization point of view, being able to find all Pareto-optimal solutions, or at least approximations to them, is therefore vital.

3 Solution Approach

3.1 Encoding of Alternatives

Alternatives to the above introduced problem are represented by a set of routes $\mathcal{R} = \{R_1, \ldots, R_m\}$, where each route R_j is driven by a particular vehicle. A route R_j defines an ordered set of customers v_i, which are visited in the sequence in which they appear in the route R_j. Overall, it is thus possible to describe the alternative as a permutation of customers, with additional partitions within the permutation defining the routes.

3.2 Constructive Phase: A Multi-objective Savings Heuristic

The construction of initial solutions is based on the savings heuristic [10]. In this procedure, each customer is first assigned to a distinct vehicle. Then, routes are, if feasible, combined, yielding a 'savings' in terms of the driven distances. For example, the routes $R_1 = \{v_1\}$ and $R_2 = \{v_2\}$ may be combined into $R_{1'} = \{v_1, v_2\}$, reducing the total driven distance by $\overline{v_0 v_1} + \overline{v_0 v_2} - \overline{v_1 v_2}$, with v_0 denoting the depot. Obviously, driving direction decisions play a role when considering time windows. Therefore, $R_{1'} = \{v_1, v_2\}$ but also $R_{2'} = \{v_2, v_1\}$ must be examined in this more general case.

It is possible to further generalize this approach for the above described two-objective case, i. e. the case in which time window violations are generally allowed but minimized by an objective function. What is needed is a mechanism controlling the degree to which time window violations are permitted. This is possible by introducing an parameter $\texttt{TARDY}_{\max}$ that acts as a constraint when executing the constructive savings procedure. While a minimum of $\texttt{TARDY}_{\max} = 0$ is possible, no trivial statements can be made about its' maximum value. Therefore, a first setting of the control parameter would be to assume $\texttt{TARDY}_{\max} = \infty$.

Algorithm 1 describes the logic behind the proposed procedure. A natural termination criterion is found when obtaining a value of $\texttt{TARDY}_{\max} < 0$, which cannot be used an a constraint in a further loop. As a result, the algorithm returns a first approximation of the Pareto-set, denoted with P^{approx}.

Algorithm 1. Multi-objective savings heuristic

1: Set $P^{approx} = \emptyset$
2: Set $\texttt{TARDY}_{\max} = \infty$
3: **repeat**
4: Create new alternative using the Clarke and Wright savings heuristic [10]: In this procedure, allow a maximum tardiness per customer of $\texttt{TARDY}_{\max}$
5: Update P^{approx} with the new alternative
6: Obtain the maximum tardiness $\texttt{TARDY}'_{\max}$ from the constructed alternative, set $\texttt{TARDY}_{\max} := \texttt{TARDY}'_{\max}$
7: Set $\texttt{TARDY}_{\max} := \texttt{TARDY}_{\max} - 1$
8: **until** $\texttt{TARDY}_{\max} < 0$
9: **return** P^{approx}

3.3 Iterative Phase: Population-Based Multi-operator Search

The elements of the initial approximation P^{approx} are further improved by means of local search. In contrast to the single-objective case, which commonly concentrates on the minimization of the traveled distances (or a similar function), some modifications must be made to heuristic search procedures in order to treat the multi-objective nature of the problem.

One the one hand, a set of solutions must be kept throughout the iterative phase. Naturally, evolutionary approaches possess this property, and therefore are a good basis for the considered case. On the other hand, the chosen neighborhood operators must be able to improve the solutions not only with respect to the classical minimization of the traveled distances, but also with respect to other criteria.

In the light of the discussion above, a population-based local search strategy making use of several neighborhood operators at once has been used to solve the problem at hand. We propose to use inversion, exchange, and move operators, each modifying the sequence of customers within the routes. An inversion is defined by the reversal of a subsequence of customers, e. g. transferring $R = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ with the given positions $p_1 = 2$ and $p_2 = 5$ into $R' = \{v_1, v_5, v_4, v_3, v_2, v_6\}$. An exchange swaps the positions of two jobs, i. e. for the example of $R = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the given positions $p_1 = 2$ and $p_2 = 5$ a resulting $R' = \{v_1, v_5, v_3, v_4, v_2, v_6\}$. Finally, a move-operator moves a customer from some position p_1 to p_2, keeping the rest of the sequence untouched.

As the neighborhoods are not disjunct, some calculations can be omitted when starting with a particular neighborhood and then continuing with another, obtaining a computational speedup by avoiding unnecessary (duplicate) moves.

Algorithm 2 describes the steps of the procedure. Again, a natural termination criterion can be mentioned, which is the identification of an approximation set P^{approx} containing local optima only.

Overall, the procedure can be characterized as a multi-point multi-neighborhood search algorithm. The concept incorporates ideas from Pareto Local Search [11] and Variable Neighborhood Search (VNS) [12]. Contrary to VNS however, all neighborhood operators are applied when improving the alternatives, thus searching the neighborhoods considerably more thoroughly.

Algorithm 2. Multi-objective local search

Require: $P^{approx}, P^{approx} \neq \emptyset$
Require: Neighborhood definitions $\mathcal{NH} = \{NH_1, \ldots, NH_k\}$
1: **repeat**
2: Select some element x from P^{approx} that is not marked as 'investigated'
3: Compute all feasible neighbors $\mathcal{NH}(x) = \bigcap_{i=1}^{k} NH_i(x)$
4: Evaluate the neighbors
5: Update P^{approx} with $\mathcal{NH}(x)$
6: **if** $x \in P^{approx}$ **then**
7: Mark x as 'investigated' w. r. t. the local search, thus excluding it in further loops from the neighborhood search procedure
8: **end if**
9: **until** All elements of P^{approx} are locally optimal w. r. t. $\mathcal{NH}$
10: **return** P^{approx}

Remark. Step 6 is required, simply because the previous line 5 may lead to a removal of x from P^{approx}.

4 Experimental Investigation

Experiments have been conducted with the initially sketched aim of investigating the computational effort that comes with solving certain data sets. Recalling, that in an interactive setting with alternating phases of search and decision making, the time allocated for the optimization runs may be limited, such insights come useful when giving recommendations for how to tackle a particular problem.

4.1 Benchmark Data

We chose to use established benchmark data from the literature for our experiments. Firstly, this data is openly available, and thus can easily be cross-checked by our peers. Secondly, different problem characteristics are present in the data sets, allowing us to compare the obtained results for instances with differing attributes.

First investigations have been carried out using the famous Solomon VRPTW instances [13]. The data sets contain 100 customers, distributed in an euclidian space, following certain distribution patterns. Such patterns comprise 'clustered', 'random', and a mixture of both, 'random-clustered' instances. Besides, variants with 'wide' and 'narrow' time windows exist, simply meaning that average values of $t_i^l - t_i^e$ are considerable smaller in the latter case.

Figure 1 plots the geographical data of the instances. Interestingly, the clustered data sets show a different geographical distribution of the customers depending on the properties of the time windows. The random and random-clustered instances are however geographically identical, independent from time window data.

4.2 Experiments and Results

Both the constructive and the iterative phase of the local search algorithm have been tested on the described benchmark data sets. A total of 56 instances has been used, as reported in more detail in Table 1.

Table 1. Solomon data sets

Graph	Time windows	No of instances
Clustered	Narrow	9
Clustered	Wide	8
Random	Narrow	12
Random	Wide	11
Random-clustered	Narrow	8
Random-clustered	Wide	8

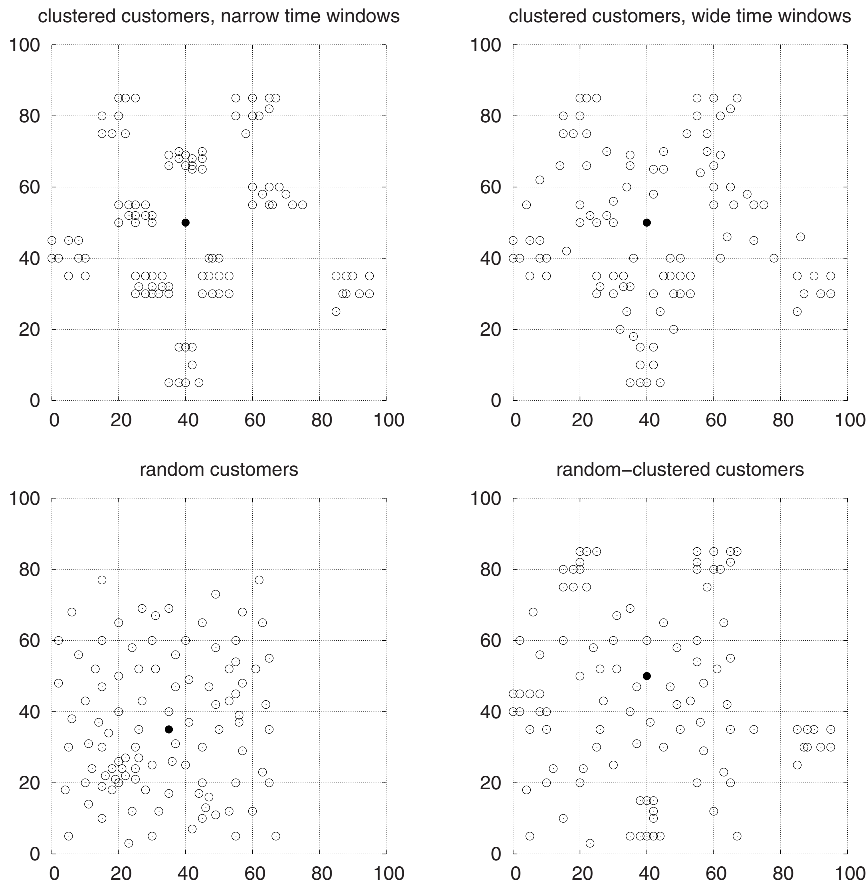

Fig. 1. Solomon instances: Geographical distribution of the depot and the customers

Constructive phase. Table 2 gives the data obtained from the constructive phases of the algorithm. Obviously, the average computational effort is heavily influenced by the underlying structures of the problem, especially when comparing the classes of clustered versus random/random-clustered data sets. Clustered data sets are in average more than ten times faster approximated than instances of the other categories.

Besides the sheer difference in terms of the amount of computations, the average number of elements in the first approximation set P^{approx} differ depending on the structural properties of the data sets. Few solutions are found for clustered instances, and significantly more for the ones of the other classes.

The two findings of this phase of the algorithm go hand in hand. It can be expected that an approximation run yielding a set of larger cardinality is more costly. Then however, this does not entirely explain the apparent difference between the classes of instances. We can suspect that the clustered instances

Table 2. Results of the constructive phase

Graph	Time windows	Average eval.	Average of $\lvert P^{approx} \rvert$	Average hypervolume
Clustered	Narrow	2,983	6	0.5775
Clustered	Wide	4,276	4	0.4724
Random	Narrow	92,702	29	0.7417
Random	Wide	44,151	31	0.6891
Random-clustered	Narrow	54,336	31	0.7378
Random-clustered	Wide	56,469	38	0.6981

favor solutions in which vehicles do not travel in between clusters, canceling out a rather big number of potential solutions. As the other, random instances do not posses this property, more savings are feasible here. In combination with the recursive computation of alternatives as given in Algorithm 1, this leads to an approximation set of larger cardinality.

The rightmost column of Table 2 states the average hypervolume [14] of the obtained approximations. By normalizing the objective function values, the hypervolume assumes a value between 0 and 1, 1 being the best-possible outcome. The reference point has been chosen such that each element of the approximation sets contributes to the hypervolume, also including the later following results of the iterative phase.

It can be seen, that the average hypervolume obtained for instances with narrow time windows is better than the one for instances with wide time windows. Besides, instances with random and random-clustered customers are better approximated when considering the resulting hypervolume indicator.

Iterative phase. The results of the succeeding improvement phase are given in Table 3.

Again, a considerable difference of the computational effort for finding local optima can be, depending on the structural properties of the instances, found. While the algorithm converges rather fast for clustered data sets, significantly more evaluations are required for random, and even more for random-clustered

Table 3. Results of the iterative phase

Graph	Time windows	Average eval.	Average of $\lvert P^{approx} \rvert$	Average hypervolume	Hypervolume improvement
Clustered	Narrow	52,473	6	0.8715	63%
Clustered	Wide	186,456	5	0.9569	178%
Random	Narrow	1,379,383	270	0.8312	13%
Random	Wide	12,425,077	418	0.8115	20%
Random-clustered	Narrow	2,032,069	320	0.8443	15%
Random-clustered	Wide	24,412,509	557	0.8497	22%

instances. Then, the influence of the time windows becomes apparent. Relatively wider time windows lead to less constrained problems, resulting in more potential candidate solutions that have to be examined before reaching a set of local optima.

The difference of the average cardinality of the approximation sets P^{approx} is rather big. This is especially the case between the clustered instances and the random/random-clustered ones with large time windows. It is interesting obtaining this result, especially as it implies that a decision making phase involving a human decision maker will have to work with candidate sets of rather different cardinality. Consequently, different Multiple Criteria Decision Making/Aiding techniques may be used, depending on whether they support decision aiding for many efficient alternatives or not.

Clearly, the iterative phase further contributes to the quality of the obtained results. In case of each single instance, significantly improved solutions have been found. Table 3 illustrates this by reporting the average hypervolume and the relative improvement upon the results of the constructive phase. Relatively higher improvements have been achieved in cases where the constructive phase first led to weaker results. This implies that the succeeding iterative phase successfully balances out the comparably weaker first approximation. Especially for clustered data sets, this is the case.

5 Conclusions

Several insights in the structures of the investigated problem and several conclusions arise from our work.

First, a fast approximation heuristic based on the savings construction procedure has been presented. A first and computationally cheap approximation has been possible by use of the proposed concept. Considerable differences of the computational effort can be found depending on the structures of the data sets. Especially for instances with geographically clustered customers, comparably few computations are required before the algorithm terminates. Nevertheless, a range of alternatives is obtained, allowing the presentation of some first variety of solutions to the decision maker.

Second, population-based local search using multiple neighborhood structures has been tested on the benchmark instances. Again, the computational effort required for finding local optima is significantly influenced by the geographical distribution of customers, and the tightness of the time windows. Maps with randomly distributed customers and wide time windows require significantly more computations than clustered data sets with narrow time windows. With respect to the overall quality of the results, the local search significantly improves the initial approximation.

With respect to our aim of providing decision support in multi-objective vehicle routing, we may conclude that clustered data sets with narrow time windows appear better suited for interactive approaches than instances with randomly distributed customers and wide time windows. In the latter case, a posteriori approaches should be given preference, assuming that the time allocated for search in between decision making phases is scarce.

References

1. Gendreau, M., Bräysy, O.: Metaheuristic approaches for the vehicle routing problem with time windows: A survey. In: MIC 2003: Proceedings of the Fifth Metaheuristics International Conference, Kyoto, Japan, August 2003, pp. 1–10 (2003)
2. Potvin, J.Y., Kervahut, T., Garcia, B.L., Rousseau, J.M.: The vehicle routing problem with time windows. Part I: Tabu search. INFORMS Journal on Computing 8(2), 158–164 (Spring 1996)
3. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows. Part II: Genetic search. INFORMS Journal on Computing 8(2), 165–172 (Spring 1996)
4. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society (52), 928–936 (2001)
5. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. European Journal of Operational Research 189(2), 293–309 (2008)
6. Park, Y.B., Koelling, C.P.: An interactive computerized algorithm for multicriteria vehicle routing problems. Computers & Industrial Engineering 16(4), 477–490 (1989)
7. Pacheco, J., Marti, R.: Tabu search for a multi-objective routing problem. Journal of the Operational Research Society (57), 29–37 (2006)
8. Phelps, S.P., Köksalan, M.: An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. Management Science 49, 1726–1738 (2003)
9. Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y.: A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 31(2), 170–186 (1997)
10. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581 (1964)
11. Paquete, L., Chiarandini, M., Stützle, T.: Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V. (eds.) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol. 535. Springer, Heidelberg (2004)
12. Hansen, P., Mladenović, N.: Variable neighborhood search. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 57, pp. 145–184. Kluwer Academic Publishers, Dordrecht (2003)
13. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time windows constraints. Operations Research 35(2), 254–265 (1987)
14. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In: Proceedings of the tenth ACM SIGEVO workshop on Foundations of Genetic Algorithms, Orlando, Florida, January 2009, pp. 87–102 (2009)