Vehicle Routing with Heterogeneous Time Windows

Petra Mutzel[1], Tim Niemann[2], Lukas Schürmann[1], Sebastian Stiller[2], and Andreas M. Tillmann[2]

[1] Universität Bonn, Institute for Computer Science, Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany
[2] Technische Universität Braunschweig, Institute for Mathematical Optimization, Universitätsplatz 2, 38106 Braunschweig, Germany

Abstract. We consider a novel variant of the heterogeneous vehicle routing problem (VRP) in which each customer has different availability time windows for every vehicle. In particular, this covers our motivating application of planning daily delivery tours for a single vehicle, where customers can be available at different times each day. The existing literature on heterogeneous VRPs typically distinguishes properties of the vehicle fleet such as costs or capacities, but apparently, windows of customers have only been regarded in a homogeneous fashion thus far. To solve the problem, we employ a branch-and-price framework based on a set partitioning formulation together with a parallelizable labeling algorithm. The heterogeneous time window structure yields notable computational gains by allowing to decompose the pricing problem as well as to utilize a customer-vehicle assignment branching rule. We show that this branching rule leads to more balanced search trees than the usual arc flow branching, and demonstrate its efficiency in numerical experiments.

Keywords: vehicle routing problem, heterogeneous time windows, branch and price

1 Introduction

The vehicle routing problem (VRP) is a well-studied problem class with a long history and numerous variants inspired by practical applications, cf., e.g., [7,2]. In the basic (capacitated) VRP, we are given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathcal{C} \cup \{v_0\}$, where $\mathcal{C}$ represents customers and v_0 is the depot of a set of vehicles $\mathcal{K}$, each with capacity Q. Each arc $e \in \mathcal{E}$ has an associated cost $c_e \geq 0$ (typically, driving time), and each customer $c \in \mathcal{C}$ has a demand q_c. We are looking for a cost-minimal set of tours in $\mathcal{G}$ that each start and end at the depot such that every customer $c \in \mathcal{C}$ is visited exactly once and the sum of all demands on each tour does not exceed the vehicle capacity.

We extend the palette of VRP variants by adding *heterogeneous time windows*: Customers can only be visited during individual availability time windows, which can be different for each vehicle. Our study of this model is motivated by a real-world practical application in which we need to schedule delivery tours for one vehicle over a given time horizon (e.g., a few weeks) and a fixed set of customers whose availability times differ from day to day. Thus, each day, the vehicle travels a different route so that at the end of the time horizon, all customers have been visited. In our application, vehicles are tantamount to days, but it is conceivable that different availability times may also directly relate to delivery or vehicle types—for instance, while recipients can be assumed to handle small packages by themselves, they may need to coordinate with helpers when receiving large pieces of furniture.

Time window constraints in VRPs have been well-known since the work of Solomon [6], but to the best of our knowledge, the case of time windows which differ for different vehicles (or days, or deliveries) has not been examined thus far. Previous works on heterogeneous VRP variants (cf. [3]) considered different aspects such as the cost and capacities on a vehicle-specific basis, but different time windows appear to be a novel concept. In particular, our approach also allows for customers to be completely unavailable for arbitrary time spans. Moreover, heterogeneous time windows not only enable addressing special customer needs, but also opens a way to reduce search tree sizes and runtime of dedicated VRP branch-and-price solvers. Notably, the heterogeneity gives rise to an effective branching rule that we shall call *vehicle assignment branching* and which compares favorably with the classical arc flow branching.

We will briefly explain our model and the problem-specific branch-and-price solution procedure in Sect. 2, including a theoretical analysis of the two branching rules. In Sect. 3, we discuss computational experiments that illustrate the practical impact of heterogeneity-based model properties and branching on different test instances. Some final remarks in Sect. 4 conclude the paper.

2 Model and Solution Method

For ease of presentation, we assume henceforth that each customer has (at most) one time window per vehicle. Nevertheless, we point out that it is possible to handle multiple (disjoint) time windows as well as, in case of daily route planning as in our motivating application, multiple vehicles per day; space limitations prevent us from providing the technical details of the corresponding modifications.

2.1 Model

Let $\mathcal{P}$ be the set of all feasible tours (w.r.t. time-window, capacity and possible further constraints), and $\mathcal{P}_k \subset \mathcal{P}$ those performed by vehicle $k \in \mathcal{K}$. For each tour p, we define its cost $C_p \coloneqq \sum_{e \in p} c_e$. With binary variables x_p to decide whether tour p partakes in the overall solution, the *set-partitioning formulation* of our VRP is then given by

$$\min \quad \sum_{p \in \mathcal{P}} C_p x_p \qquad \text{(SPF)}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}: i \in p} x_p = 1 \quad \forall i \in \mathcal{C}, \quad \sum_{p \in \mathcal{P}_k} x_p = 1 \quad \forall k \in \mathcal{K}, \; x_p \in \{0,1\} \quad \forall p \in \mathcal{P}\,.$$

The first constraint guarantees that each customer is visited exactly once, and the second one that each vehicle is used on at most one tour (for technical reasons, we include empty tours with cost 0 in each $\mathcal{P}_k$ to write this in equality form); the latter stems from our daily route planning application and may take a different form in other cases.

Due to the exponential number of tours and the resulting impracticality of directly solving (SPF), we adopt the established VRP solution approach of *branch-and-price* with a dedicated labeling algorithm; see, e.g., [2].

2.2 Pricing Problem

The general idea of branch-and-price is to start with few tours $\mathcal{P}' \subset \mathcal{P}$, and then iteratively add new tours to $\mathcal{P}'$ and re-optimize the reduced problem (with variables x_p, $p \in \mathcal{P}'$) until a provably optimal solution to the original problem is found. To find promising new tours $p \in \mathcal{P} \setminus \mathcal{P}'$, we consider the following *reduced-cost pricing problem*: Given an optimal dual solution (π^*, τ^*) for the LP relaxation of the reduced problem, solve

$$r(\pi^*, \tau^*) = \min_{k \in \mathcal{K}} \Big\{ -\tau_k^* + \min_{p \in \mathcal{P}_k} \sum_{(i,j) \in p} c_{ij} - \pi_i^* \Big\}\,. \tag{RCPP}$$

Since the depot node $v_0 \notin \mathcal{C}$ has no dual variable, we define $\pi_{v_0}^* := 0$ to simplify notation. If we find a tour with negative reduced cost ($r(\pi^*, \tau^*) < 0$), we add it to $\mathcal{P}'$; otherwise, the reduced LP was solved to optimality (w.r.t. all tours, including omitted ones).

This pricing problem amounts to an *elementary shortest path problem with resource constraints*, which is strongly NP-hard and usually solved using a labeling algorithm, cf. [2]. Due to space limitations, we omit the relatively straightforward details on adapting the labeling scheme to our present setting, but note that vehicle capacities and customer time windows are translated to resource constraints. Since (RCPP) naturally decomposes into independent pricing subproblems for every vehicle $k \in K$, we only need to consider the customers who have a time window for the corresponding vehicle $k \in K$ for each subproblem. Depending on the instance, this significantly reduces the size of the relevant subgraph, which in turn speeds up the labeling algorithm. Furthermore, the subproblems can be solved in parallel.

2.3 Branching Rules

In a branch-and-price framework, branching on the standard problem variables often interferes with the pricing mechanism. Several alternative branching rules for VRPs have been proposed to deal with this issue, see [2] for an overview of the most common ones. Branching on *arc flow variables* $y_e := \sum_{p \in \mathcal{P}: e \in p} x_p$, $e \in \mathcal{E}$,

Table 1. Maximal number of branchings for different branching rules

branching variables	number of 0-branches	number of 1-branches
tour variables (x_p)	$\mathcal{O}(2^{\lvert\mathcal{V}\rvert})$	$\lvert\mathcal{K}\rvert$
arc flow variables (y_e)	$\lvert\mathcal{E}\rvert - \lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}\rvert$
vehicle assignment variables (d_i^k)	$(\lvert\mathcal{K}\rvert - 1)\lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}\rvert$

appears to be the most popular strategy, even though the branching decisions, especially setting $y_e = 0$ (removing an arc), have a rather low overall impact.

Our heterogeneous structure allows us adapt the branching idea originally introduced by Ryan and Foster in [4], and formulated in [5] for the general assignment problem. The idea is to branch on assignment decisions; in the present setting, these translate to decisions whether to let specific customers get served by certain vehicles. To that end, we define auxiliary *vehicle assignment variables*

$$d_i^k \coloneqq \sum_{p \in \mathcal{P}_k : i \in p} x_p \in [0,1], \quad \forall\, i \in \mathcal{C}, k \in \mathcal{K}\,.$$

Then, we can branch on a (fractional) d_i^k, setting it to 0 or 1, respectively. These *vehicle assignment branching* decisions can easily be accounted for in subsequent pricing steps: $d_i^k = 0$ leads to the exclusion of customer i from pricing subproblem for vehicle k, and $d_i^k = 1$ removes i from all pricing subproblems except for vehicle k. Moreover, any exact pricing scheme can be adapted to ensure the condition $d_i^k = 1$ remains valid, cf. [5], and in leaves of the resulting decision tree, there exists an integral optimal solution for the LP relaxation of (SPF). The latter arises from the fact that for each solution x such that $d_i^k \in \{0,1\}$ for all $i \in \mathcal{C}, k \in \mathcal{K}$, all tours in $\mathcal{P}_k^{>0} \coloneqq \{p \in \mathcal{P}_k \,:\, x_p > 0\}$ visit the same set of customers for each $k \in \mathcal{K}$, as $\sum_{p \in \mathcal{P}_k} x_p = 1$ must be satisfied. Consequently, the C_p values are identical for each $p \in \mathcal{P}_k^{>0}$, and any solution with $x_{p'} = 1$ for an arbitrary $p' \in \mathcal{P}_k^{>0}$ and $x_p = 0$ for each $p \in \mathcal{P}_k^{>0} \setminus \{p'\}$ is feasible and optimal. Applying this transformation for every $k \in \mathcal{K}$ yields an integral optimal solution. We remark that the variables d_i^k are not actually included in the problem, but are handled implicitly during pricing.

Table 1, which gives the potential branching tree depths for all previously mentioned branching rules, exhibits a theoretical advantage of our rule. Clearly, vehicle assignment branching leads to more balanced and possibly much smaller tree, especially when the instance has many arcs but relatively few vehicles, as in this case $|\mathcal{K}| \times |\mathcal{V}| \ll |\mathcal{E}|$. The practical impact of our branching rule will be assessed by means of the computational experiments discussed in the following.

3 Experimental Results

To obtain test instances for our VRP variant, we modified Solomon's data set [6], which includes only homogeneous time windows, for our purposes. Based on the three instances R101, C101 and RC101, we generated new instances by taking the first $|\mathcal{C}| \in \{40, 50, \ldots, 100\}$ customers and inheriting travel and service times, customer demands and vehicle capacities. We then add a time window for each customer-vehicle pair $i \in \mathcal{C}, k \in \mathcal{K}$ with probability $\alpha_{\text{TW}} \in \{0.25, 0.5, 0.75, 1\}$, retaining the lengths of time windows from the respective base instance but randomly choosing their starting times (such that all time windows lie within depot working hours), where the size of $\mathcal{K}$ was set heuristically to ensure feasibility.

We implemented our approach in `C++` with SCIP 8.0 [1] and tested it on a 24-core machine; on average, parallelization of the pricing subproblems saved about half the computation time with this setup, compared to single-thread execution. We solved each instance twice: once with our vehicle assignment branching, and once using arc flow branching. In both cases, we chose the variable with value closest to 0.5 for branching. The results in Table 2 focus on the instances based on R101; those based on the other two base instances yielded similar results. Besides instance parameters, we report the runtime in seconds ("DNF" marks cases for which the solution process did not finish after one hour), number of branching nodes, and optimality gap at termination for both variants. We also state the best objective value found by the vehicle assignment branching variant; the objective value for the arc flow branching variant was mostly very similar, so differences in gap values are predominantly due to dual bounds. All decimals were rounded to two significant digits.

From Table 2, we can make some key observations: Almost all instances can be solved faster and with fewer search nodes using vehicle assignment branching than with arc flow branching, especially those with many time windows per customer (larger α). For instances with many customers, the vehicle assignment branching variant can solve more instances and often significantly reduce the final gap when hitting the time limit. In general, solving times increase notably with the number of time windows; for example, with 70 customers, the instance with $\alpha_{\text{TW}} = 0.25$ can be solved in well under a second, whereas more than an hour is needed if every customer has a (different) time window for each vehicle. However, increasing customer flexibility naturally allows for more efficient routings, as reflected by the primal bounds decreasing significantly the larger α_{TW} gets.

Finally, it is worth mentioning another empirical observation that is not apparent from Table 2: If a lot of customer-vehicle pairs have similar time windows, i.e., the instance "approaches" time-window homogeneity, then arc flow branching is preferable to vehicle assignment branching, which can likely be explained by the former then affecting several vehicles simultaneously.

4 Concluding Remarks

We introduced the concept of heterogeneous time windows for the VRP to account for modern-day customer availability requirements, and demonstrated that it allows for often significant computational gains by decomposition of pricing problems and the specialized vehicle assignment branching rule. In ongoing work, we also successfully adapted our approach to multiple time windows per day in

Table 2. Results for test instances based on R101, comparing two branching rules

			vehicle assignment branching				arc flow branching		
$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{K}\rvert$	α_{TW}	best obj.	time	nodes	% gap	time	nodes	% gap
40	11	0.25	1231.45	0.01	1	0	0.01	1	0
40	8	0.50	1043.80	0.05	3	0	0.05	3	0
40	7	0.75	811.64	0.96	37	0	1.19	59	0
40	12	1.00	741.23	6.79	573	0	12.44	1239	0
50	13	0.25	1604.27	0.01	1	0	0.01	1	0
50	10	0.50	1271.13	0.06	1	0	0.06	1	0
50	14	0.75	920.46	2.23	129	0	2.10	103	0
50	12	1.00	853.98	14.28	265	0	14.80	299	0
60	14	0.25	1724.36	0.08	13	0	0.13	25	0
60	17	0.50	1119.48	0.54	43	0	1.16	70	0
60	14	0.75	983.33	49.05	1490	0	125.34	4718	0
60	15	1.00	943.59	1590.46	20503	0	DNF	69128	0.71
70	16	0.25	2080.74	0.03	1	0	0.03	1	0
70	17	0.50	1335.39	23.85	1286	0	67.52	4251	0
70	17	0.75	1174.52	3527.82	51265	0	DNF	70380	1.15
70	16	1.00	1096.63	DNF	7718	1.17	DNF	7294	8.35
80	18	0.25	2213.67	3.65	265	0	12.33	1145	0
80	21	0.50	1418.48	485.65	23473	0	2435.35	114433	0
80	20	0.75	1221.76	DNF	26970	0.3	DNF	36499	1.27
80	18	1.00	1148.86	DNF	2609	0.77	DNF	3010	0.91
90	25	0.25	2024.82	68.85	8177	0	396.55	47164	0
90	21	0.50	1534.87	DNF	68465	1.4	DNF	77536	2.21
90	20	0.75	1602.31	DNF	8807	21.37	DNF	8744	21.86
90	19	1.00	1504.87	DNF	360	24.71	DNF	397	24.80
100	25	0.25	2124.97	263.48	18639	0	DNF	218423	0.76
100	23	0.50	1540.94	508.18	6227	0	DNF	54166	0.57
100	19	0.75	1570.67	DNF	2325	13.67	DNF	3637	13.69
100	19	1.00	1581.19	DNF	16	25.89	DNF	15	25.97

a daily delivery tour planning application. As future research, we plan to assess the algorithmic efficiency in yet more general variants with, e.g., multiple vehicles per day and incorporating robustness w.r.t. travel delays. Furthermore, it is of interest to construct problem-specific primal and pricing heuristics to further speed up computations and enable solving larger or more complicated instances.

Acknowledgments. The authors gratefully acknowledge funding by the German Federal Ministry for Education and Research, grant nos. 05M20MBA-LeoPlan and 05M20PDA-LeoPlan.

References

1. Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S.J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M.E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., Witzig, J.: The SCIP Optimization Suite 8.0. ZIB-Report 21-41, Zuse Institute Berlin (2021). http://nbn-resolving.de/urn:nbn:de:0297-zib-85309
2. Costa, L., Contardo, C., Desaulniers, G.: Exact Branch-Price-and-Cut Algorithms for Vehicle Routing. Transp. Sci. 53(4), pp. 946–985 (2019). doi:10.1287/trsc.2018.0878
3. Koç, Ç., Bektaş, T., Jabali, O., Laporte, G.: Thirty years of heterogeneous vehicle routing. Eur. J. Oper. Res. 249(1), pp. 1–21 (2016). doi:10.1016/j.ejor.2015.07.020
4. Ryan, D.M., Foster, B.A.: An Integer Programming Approach to Scheduling. In: Wren, A. (ed.) Computer Scheduling of Public Transport, pp. 269–280. North Holland, Amsterdam (1981).
5. Sol, M.: Column Generation Techniques for Pickup and Delivery Problems. PhD Thesis, Technische Universiteit Eindhoven, The Netherlands (1994).
6. Solomon, M.M.: Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms (Heuristics). PhD Thesis, University of Pennsylvania, PA, USA (1984).
7. Vidal, T., Laporte, G., Matl, P.: A concise guide to existing and emerging vehicle routing problem variants. Eur. J. Oper. Res. 286(2), pp. 401–416 (2020). doi:10.1016/j.ejor.2019.10.010