



# Preprocessing to Reduce Vehicle Capacity for Routing Problems

Twan Dollevoet<sup>1</sup> · Remy Spliet<sup>1</sup>

Received: 20 April 2022 / Accepted: 10 April 2023 / Published online: 15 May 2023

© The Author(s) 2023

## Abstract

We provide a preprocessing method to reduce the vehicle capacity for instances of the capacitated vehicle routing problem. This improves the LP bound of many formulations of the capacitated vehicle routing problem. It also speeds up common algorithms for which the computation time depends on the vehicle capacity. Our simulation experiments suggest that, perhaps surprisingly, often the vehicle capacity is very tight in the sense that it cannot be reduced by much.

**Keywords** Vehicle routing · Capacity · Preprocessing

## 1 Introduction

Vehicle routing is a term that is used in the context of Operations Research to refer to a variety of optimization problems. These problems have in common that they are defined on a graph which is traversed by one or multiple vehicles of finite capacity to supply demand of customers. There are many variants, many real-world applications, and many exact and heuristic algorithms, see, e.g., [1].

Most exact algorithms, and even some heuristics, rely on LP bounds of a mixed integer programming formulation. Moreover, some of the fastest algorithms for computing LP bounds use subroutines that require pseudo-polynomial computation time. In particular, their computation time is polynomial in the vehicle capacity, like in the case of [2] and [3]. Our contribution is a preprocessing method to reduce the vehicle capacity, which improves the LP bounds and reduces the computation time of the aforementioned algorithms.

In this paper, we focus specifically on the capacitated vehicle routing problem (CVRP) although the results carry over to other vehicle routing problems. The CVRP is defined on a complete digraph  $G = (N, A)$ , where  $N = \{0\} \cup N'$ , 0 represents the depot and  $N'$  the set of customers. An unlimited number of vehicles of finite capacity  $Q$

---

✉ Twan Dollevoet  
dollevoet@ese.eur.nl

<sup>1</sup> Econometric Institute, Erasmus University Rotterdam, Rotterdam, The Netherlands

are used to satisfy demand. The demand of customer  $i \in N'$  is denoted by  $q_i$ , satisfying  $0 \leq q_i \leq Q$ . A route is a simple cycle in  $G$  which starts and ends at the depot, such that the total demand of the visited nodes does not exceed the vehicle capacity. A cost  $c_{ij}$  is incurred for traversing an arc  $(i, j) \in A$ . The CVRP is to find a collection of routes such that all demands are satisfied at minimum cost.

In Section 2, we demonstrate the preprocessing method. Here we also provide an example of an instance for the CVRP and a well-known formulation for which the LP bound improves after preprocessing. Next, we indicate in Section 3 that for none of the 233 considered well-known benchmark instances of the CVRP, the capacity can be reduced using our preprocessing method. We then show a series of simulations which perhaps surprisingly suggest that in many cases the vehicle capacity can only be reduced by a very small amount, if at all. We comment on the implications of this in Section 4.

## 2 Preprocessing

Next we provide our preprocessing method and discuss the computational gains it provides. Observe that there might not exist a subset of customers for which the total demand is exactly the vehicle capacity. Determining whether this is the case is precisely a SUBSET-SUM-DECISION problem, one of the 21 NP-complete problems of [4] which was referred to as the KNAPSACK problem. We denote by  $Q'$  the maximum total demand that fits in a vehicle amongst subsets of customers, i.e.,

$$Q' = \max_{S \subseteq N'} \left\{ \sum_{i \in S} q_i : \sum_{i \in S} q_i \leq Q \right\}.$$

Finding  $Q'$  is a SUBSET-SUM-OPTIMIZATION problem, which first appeared in [5] with the name VALUE-INDEPENDENT KNAPSACK problem, and should not be confused with the SUBSET-SUM-DECISION problem. Solving the SUBSET-SUM-OPTIMIZATION problem is weakly NP-hard, since SUBSET-SUM-DECISION reduces to it and pseudo-polynomial time algorithms exist to solve it.

Modifying an instance of the CVRP by replacing  $Q$  with  $Q'$  does not alter the set of feasible solutions. Moreover, this can be done as a preprocessing method. Next, we show that there are computational gains from doing this.

We provide an instance for which the LP bound improves after replacing  $Q$  with  $Q'$  for the well-known two-index vehicle flow formulation (2ICF) of the CVRP by [6]. Denoting by  $x_{ij}$  the decision variable indicating whether arc  $(i, j) \in A$  is used, 2ICF is

$$\begin{aligned}
 & \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 & \sum_{j \in N} x_{ji} = 1 \quad \forall i \in N' \\
 & \sum_{j \in N} x_{ij} = 1 \quad \forall i \in N' \\
 & \sum_{i \in S, j \in N \setminus S} x_{ij} \geq \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil \quad \forall S \subseteq N' \\
 & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A.
 \end{aligned} \tag{1}$$

The last set of constraints in Eq. (1) are known as rounded capacity inequalities, and there are exponentially many. They are strongly NP-hard to separate, see [7], and heuristics like those of [8] are commonly used for this purpose. Observe that for all sets of customers that fit in one vehicle, i.e.,  $S \subseteq N'$  for which  $\sum_{i \in S} q_i \leq Q$ , the corresponding rounded capacity cut is not affected by replacing  $Q$  with  $Q'$  since  $\left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil = 1 = \left\lceil \frac{\sum_{i \in S} q_i}{Q'} \right\rceil$ . Nevertheless, for the following instance the LP bound improves from replacing  $Q$  with  $Q'$ .

Consider an instance with three customers. Let  $c_{0i} = c_{i0} = 5$  and  $c_{ij} = 8$  for all  $i, j \in N'$ . Let  $q_i = 2$  for all  $i \in N'$  and  $Q = 3$ . Note that  $Q' = 2$  for this instance, and replacing  $Q$  with  $Q'$  affects only a single constraint of 2ICF. In particular the rounded capacity cut  $x_{10} + x_{20} + x_{30} \geq 2$  becomes  $x_{10} + x_{20} + x_{30} \geq 3$ . It is easily verified that  $x_{0i} = x_{i0} = \frac{2}{3}$  and  $x_{ij} = \frac{1}{6}$ , for all  $i, j \in N'$ , is a feasible solution to the LP relaxation of 2ICF, with objective value 28. In fact it is optimal. When replacing  $Q$  with  $Q'$ , there is only one feasible solution to the LP relaxation of 2ICF, namely  $x_{0i} = x_{i0} = 1$  for all  $i \in N'$  with objective value 30, which is therefore optimal. Coincidentally, this is even the optimal integer solution. We conclude that the LP bound has improved from replacing  $Q$  with  $Q'$ .

While an improved LP bound might provide computational gains in a branching algorithm, next we discuss a more direct computational gain. Consider the set partitioning formulation, which is one of the strongest known CVRP formulations, see [9]. Denoting by  $R$  the set of all routes,  $x_r$  the decision variable indicating whether route  $r \in R$  is selected,  $c_r$  the cost of  $r \in R$ , and  $a_{ir}$  indicating whether  $i \in N'$  is visited on  $r \in R$ , the set partitioning formulation is

$$\begin{aligned}
 & \min \sum_{r \in R} c_r x_r \\
 & \sum_{r \in R} a_{ir} x_r = 1 \quad \forall i \in N' \\
 & x_r \in \{0, 1\} \quad \forall r \in R.
 \end{aligned}$$

First note that the set partitioning formulation is not affected by replacing  $Q$  with  $Q'$ . Indeed, the set  $R$  remains the same, and as a result so does the LP bound. However, fast algorithms do not use this formulation, for which an explanation might be

that computing the set partitioning bound of the CVRP is NP-hard, see [10]. Instead, the set  $R$  is extended with routes that visit customers more than once, allowing to compute the LP bound in pseudo-polynomial time. Two examples are  $q$ -routes, see [2], and ng-routes, see [11], which allow for computation time that is polynomial in the vehicle capacity. Hence, the computation time goes down when replacing  $Q$  with  $Q'$ . Of course, also in this case the LP bound might improve, since the extended set corresponding to  $Q'$  might be smaller.

### 3 Experiments

To assess the computational gains of our preprocessing procedure, we consider the following sets of benchmark instances of the CVRP: A, B, CMT, E, F, M, P, Tai, X by [12–16] and [17], which can all be conveniently downloaded from [18]. Out of the 14 CMT instances we only consider 7 because they come in pairs with the same demand. In total, these are 233 instances.

For each instance we compute  $Q'$  by using our own implementation of a standard dynamic programming algorithm for the SUBSET-SUM-OPTIMIZATION problem from [19]. For none of the 233 instances does  $Q'$  differ from  $Q$ . The question is now whether there is something particular about these benchmark instances, or whether  $Q'$  and  $Q$  are generically not much different. Next, we show simulation experiments that perhaps surprisingly suggest that the latter seems to be the case.

Consider the following simulation experiment. For a given number of customers, we randomly generate demand. In the benchmark instances demand is always integer. Although we deviate from this, note that given the finite precision of modern computers, we can effectively only sample from discrete distributions. That is, we consider a discrete support, resulting in only a finite number of different demand realizations. Given  $Q$ , we then compute  $Q'$  and assess their difference.

The demand distribution affects the difference between  $Q'$  and  $Q$ . Roughly stated, for a fixed number of customers, the higher the cardinality of the support the less likely it becomes that a subset of demands sums precisely to  $Q$ . Furthermore, this becomes more likely when the number of customers increases. Next, we evaluate the magnitude of these effects in a simulation experiment.

In Table 1, the results are shown of replicating the simulation experiment 100,000 times. In each replication, demand for 30 customers is generated using a discrete uniform distribution between 1 and 30, where we vary the decimal precision from 0 to 6 decimals. This means that for 0 decimals the support has cardinality 30, and for 6 decimals the support has cardinality 29,000,001. We use a vehicle capacity of 100. For the case of 0 decimals, this resembles the demand generating process of the A and B instances, although for those instances 10% of the demands are multiplied by 3.

The column ‘# Dec’ in Table 1 indicates the number of decimals considered. The column ‘# Diff’ indicates the number of replications out of 100,000 for which  $Q'$  and  $Q$  are different. The columns ‘Min. Diff’, ‘Avg. Diff’ and ‘Max. Diff’ show the minimum, average and maximum difference between  $Q'$  and  $Q$  respectively, only considering those replications for which a difference was observed. Note that the

**Table 1** Difference between  $Q'$  and  $Q$ , 100,000 replications, 30 customers, 100 capacity, discrete uniform demand between 1 and 30

# Dec	# Diff	Min. Diff	Avg. Diff	Max. Diff
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	196	0.000100	0.0001087	0.000300
5	15,624	0.000010	0.0000186	0.000370
6	71,513	0.000001	0.0000075	0.000359

minimum and maximal difference can be computed with absolute precision and are displayed as such, while for the average difference this is not the case and we report only relevant decimal places.

Observe from Table 1 that there are no differences for any replications with a precision of up to three decimals. With a precision of 4 decimals or more, the number of differences increases. These results support the claim that it becomes more likely to observe a difference between  $Q'$  and  $Q$  as the cardinality of the support grows. Furthermore, note that in none of the 700,000 cases the vehicle capacity can be reduced by more than 0.0004%. We consider this a rather small difference, especially taking into account that only 30 customers are used. We finally remark that for a precision of 4 decimal places and higher, the minimum difference is always the smallest possible difference given the precision.

We repeat this experiment with a different demand distribution and more customers. The demands in the Tai instances are generated using an exponential distribution, with realizations rounded to the nearest integer and truncated at the vehicle capacity. The smallest Tai instance has 75 customers. To resemble this case, we generate exponentially distributed demand with a mean of 175 for 75 customers, with realizations rounded to 0 up to 5 decimal places and truncated at the vehicle capacity. We use a vehicle capacity of 1500, and again replicate each experiment 100,000 times. In none of these 600,000 new cases is a difference observed.

## 4 Conclusion

In this paper, a preprocessing method is provided which reduces the vehicle capacity of a CVRP instance. This provides computational gains by improving LP bounds of mixed integer programming formulations of the CVRP, and decreases computation time for algorithms of which the computation time depends on vehicle capacity.

Although we demonstrate these computational gains with concrete examples, in none of the 233 considered benchmark instances is there any gain from preprocessing. Moreover, our simulation experiments suggest that in many cases, we should not expect a big difference between the vehicle capacity and the preprocessed capacity. This would imply that generically we might not expect much direct computational gains from the preprocessing method.

However, the insights provided in this paper might also prove beneficial in future analysis and algorithmic development. We claim that we may now assume that vehicle capacity is tight, in the sense that there exists at least one subset of customers of which the total demand is precisely the capacity. Either this is immediately the case, or an instance can be modified to make it so in pseudo-polynomial time.

The preprocessing method straightforwardly carries over to other vehicle routing problems. In future research, the preprocessing method can be refined further for such problems. For example, for the vehicle routing problem with time windows, one can consider only those subsets of customers for which a feasible route exists satisfying all time windows. This might provide larger reductions of the vehicle capacity, or a reduction in more cases, potentially leading to more computational gains.

**Funding** The author(s) received no financial support for the research, authorship, and/or publication of this article.

**Data Availability** The data that is used in the experiments can be obtained from [18] or generated as described in the manuscript.

## Declarations

**Conflict of Interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Toth P, Vigo D (2014) *Vehicle Routing, Problems, Methods, and Applications*. (Society for Industrial and Applied mathematics), 2nd
2. Fukasawa R, Longo H, Lysgaard J, Poggi de Aragão M, Reis M, Uchoa E, Werneck R (2006) Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Math Program* 106(3):491–511
3. Pecin D, Pessoa A, Poggi M, Uchoa E (2017) Improved branch-cut-and-price for capacitated vehicle routing. *Math Program Comput* 9(1):61–101
4. Karp R (1972) Reducibility among combinatorial problems. *Complexity of Computer Computations* (Springer, Boston) 85–103
5. Faaland B (1973) Solution of the Value-Independent Knapsack Problem by Partitioning. *Oper Res* 2(1):332–337
6. Laporte G, Nobert Y (1987) Exact algorithms for the vehicle routing problem. *North-Holland Mathematics Studies* 132:147–184
7. Diarrassouba I (2017) On the complexity of the separation problem of rounded capacity inequalities. *Discret Optim* 25:86–104

8. Lysgaard J, Letchford A, Eglese R (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math Program* 100(2):423–445
9. Letchford A, Salazar-González J (2015) Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem. *Eur J Oper Res* 244(3):730–738
10. Dollevoet T, Munari P, Spliet, R (2020) A p-step formulation of the capacitated vehicle routing problem. *Econometric Institute Report Series EI-2020-01*, Erasmus University Rotterdam
11. Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Oper Res* 59(5):1269–1283
12. Augerat P (1995) Approche polyédrale du problème de tournées de véhicules. PhD thesis Institut Nationale Polytechnique de Grenoble
13. Christofides N, Eilon S (1969) An Algorithm for the Vehicle Dispatching Problem. *Journal of the Operational Research Society* 20(3):309–318
14. Christofides N, Mingozzi A, Toth, P (1979) The vehicle routing problem. *Combinatorial Optimization*, Wiley, Chichester:315–338
15. Fischer M (1994) Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. *Oper Res* 42(4):626–642
16. Rochat Y, Taillard E (1995) Probabilistic diversification and intensification in local search for vehicle routing. *J Heuristics* 1(1):147–167
17. Uchoa E, Pecin D, Pessoa A, Poggi M, Vidal T, Subramanian A (2017) New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur J Oper Res* 257(3):845–858
18. CVRPLIB [vrp.atd-lab.inf.puc-rio.br](http://vrp.atd-lab.inf.puc-rio.br). Accessed April 2022
19. Kellerer H, Pferschy U, Pisinger D (2003) *Knapsack Problems*. Springer-Verlag, Berlin Heidelberg

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.