10,000 optimal CVRP solutions for testing machine learning based heuristics

Eduardo Queiroga[1], Ruslan Sadykov[1], Eduardo Uchoa[2], Thibaut Vidal[3,4]

[1] INRIA Bordeaux-Sud-Ouest, France
[2] Universidade Federal Fluminense, Brazil
[3] CIRRELT & SCALE-AI Chair in Data-Driven Supply Chains, MAGI, Polytechnique Montréal, Canada
[4] Department of Computer Science, Pontifical Catholic University of Rio de Janeiro, Brazil
eduardo.vieira-queiroga@inria.fr, ruslan.sadykov@inria.fr, eduardo_uchoa@id.uff.br, thibaut.vidal@polymtl.ca

Abstract

We introduce a benchmark of 10,000 instances with heterogeneous characteristics for the capacitated vehicle routing problem. We also provide optimal solutions for almost all of them along with a generator to produce additional training and validation data. This benchmark aims to permit a more systematic comparison of machine learning based search algorithms on this important problem. We also emit recommendations regarding the correct use of this dataset.

Introduction

Vehicle routing is one of the most studied class of combinatorial optimization problems due to its difficulty and practical impact (Vidal, Laporte, and Matl 2020). The capacitated vehicle routing problem (CVRP), in particular, seeks minimum-distance itineraries to visit a set of clients subject to constraints on truck capacities. Whereas early studies on this topic come from the operations research (OR) domain (Dantzig and Ramser 1959), there has recently been a significant regain of interest on it in machine learning (ML), with the promise of bringing major improvements to existing search techniques or even new search paradigms. In OR and ML likewise, new algorithm developments are guided by empirical evaluations which permit to evaluate the usefulness of different search techniques.

The importance of rigorous empirical evaluation of algorithms has been extensively recognized (Johnson 1999). Empirical evaluations often serve as a guide to theoretical studies, and lead to search strategies that are efficient in practice. For decades, the progress on search heuristics has led to solutions that are increasingly closer to the best possible (i.e., optimal solutions) for the CVRP. As of today, *State-Of-the-Art* (SOA) heuristics (Christiaens and Vanden Berghe 2020; Vidal 2021; Máximo and Nascimento 2021; Accorsi and Vigo 2021) typically compete to bring improvements of distance of about 0.5% on common test data. Whereas such differences may look small at first glimpse, they nonetheless represent significant transportation costs, and also guide methodological development towards better search concepts that are applicable to a wide range of combinatorial optimization problems. However, as the differences become increasingly small, and with the current pressure towards *positive* or SOA results in academic publications, there exists a selection bias which limits the statistical significance of most studies (Ioannidis 2005). In view of this, it is critically important to enforce rigorous practices for algorithmic evaluations to discern significant effects from noise.

There exists a number of important guidelines regarding experimental evaluations, many of them are discussed in (Accorsi, Lodi, and Vigo 2021). In particular, an important factor is the choice of a *common benchmark* for comparing solution methods. On this aspect, OR and ML methods strongly differ. In the OR domain, comparisons on a limited set of common instances are a standard for algorithmic evaluations. This permitted an uninterrupted stream of algorithmic comparisons during decades. The datasets used to that end have evolved in their complexity over the years, initially counting a few dozens of clients, and as of lately including between 100 and 1000 customers with varying characteristics (Uchoa et al. 2017), or even more (Arnold, Gendreau, and Sörensen 2019).

Whereas the number of instances used in the OR domain is relatively limited, most machine learning methodologies (Nazari et al. 2018; Chen and Tian 2019; Kool, van Hoof, and Welling 2019; Hottung and Tierney 2019; Delarue, Anderson, and Tjandraatmadja 2020; Kool et al. 2021; Hottung, Kwon, and Tierney 2021; Xin et al. 2021) require training and evaluation over a large number of instances to be meaningful. The existing instances from the OR domain have been insufficient for this purpose. As a consequence, due to heterogeneous requirements in terms of data between different ML methods, common benchmark and not systematically used. In (Kool, van Hoof, and Welling 2019) for example, the test dataset is drawn from the same distribution but different from (Nazari et al. 2018). This limits the ability to compare different methods with a good degree of precision, due to interacting factors and uncertainties that arise during data generation.

Generation of the 10,000 dataset

The new dataset follows a similar generation scheme to (Uchoa et al. 2017) and produces 2D Euclidean instances with the following configurations:

- *Depot positioning*: **1.** Random; **2.** Centered; **3.** Cornered

- *Customer positioning*: **1.** Random; **2.** Clustered; **3.** Random-Clustered
- *Demand distribution*: **1.** Unitary; **2.** Small values with large CV (Coefficient of Variation); **3.** Small values with small CV; **4.** Large values with large CV; **5.** Large values with small CV; **6.** Depending on quadrant; **7.** Many small values and few large values
- *Average route size* r: **1.** very short, r from $U[3,5]$; **2.** short, r from $U[5,8]$; **3.** medium, r from $U[8,12]$; **4.** long, r from $U[12,16]$; **5.** very long, r from $U[16,25]$; **6.** ultra long, r from $U[25,50]$

The process differs from the original because it simplifies the distributions of r, which is no longer taken from a continuous triangular distribution. Furthermore, we introduce the sixth level (ultra long) for r. The Cartesian product of all the sets of configurations produces 378 instance groups. To reach 10,000 instances with 100 customers, we generated 27 instances for each of the first 172 groups (considering a lexicographic order of the groups), and 26 instances for each of the remaining 206 groups. We managed to find optimal solutions for almost all the new dataset using SOA branch-cut-and-price algorithm (Pessoa et al. 2020), including customized parameterizations for some specific groups, and branch-and-cut algorithm (Lysgaard, Letchford, and Eglese 2004) which performs well for ultra long routes.

Suggested Experimental Guidelines

1. The instance generator, provided in Python, can be used for generating as many training instances as desired.
2. One of the publicly available state-of-the-art heuristics can be used for generating very good solutions for the training instances (using the known exact methods for solving hundreds of thousands instances would be too time consuming).
3. The 10,000 instances dataset can be used for the final testing.

The potential advantages of following that guideline:

- The provided generator is the same used for creating the X instances (Uchoa et al. 2017). It was carefully designed to create that very diversified dataset, mimicking the features of real-world problems. The X instances were already used in hundreds of published works and are now the most widely used CVRP benchmark. It is desirable for the community to not go back into more simplistic ways of generating instances.
- It is recommended to have different methods tested over exactly the same instances, not only instances that are generated in a similar way. That point was strongly advised in (Johnson 1999) as a way of eliminating a source of noise in the comparisons. The size of the testing dataset is big enough to produce statistically significant results and to make overfitting unlikely (the 10,000 dataset should not be used for training!).
- Finally, the existence of optimal solution values for the 10,000 dataset allows measuring *absolute errors*, which is certainly better than measuring relative errors with respect to a reference method.

References

Accorsi, L.; Lodi, A.; and Vigo, D. 2021. Guidelines for the Computational Testing of Machine Learning approaches to Vehicle Routing Problems. arXiv:2109.13983.

Accorsi, L.; and Vigo, D. 2021. A Fast and Scalable Heuristic for the Solution of Large-Scale Capacitated Vehicle Routing Problems. *Transport. Sci.*, 55(4): 832–856.

Arnold, F.; Gendreau, M.; and Sörensen, K. 2019. Efficiently solving very large-scale routing problems. *Comput. Oper. Res.*, 107: 32–42.

Chen, X.; and Tian, Y. 2019. Learning to Perform Local Rewriting for Combinatorial Optimization. arXiv:1810.00337.

Christiaens, J.; and Vanden Berghe, G. 2020. Slack Induction by String Removals for Vehicle Routing Problems. *Transport. Sci.*, 54(2): 417–433.

Dantzig, G.; and Ramser, J. 1959. The truck dispatching problem. *Manage. Sci.*, 6(1): 80–91.

Delarue, A.; Anderson, R.; and Tjandraatmadja, C. 2020. Reinforcement Learning with Combinatorial Actions: An Application to Vehicle Routing. arXiv:2010.12001.

Hottung, A.; Kwon, Y.-D.; and Tierney, K. 2021. Efficient Active Search for Combinatorial Optimization Problems. arXiv:2106.05126.

Hottung, A.; and Tierney, K. 2019. Neural large neighborhood search for the capacitated vehicle routing problem. *arXiv:1911.09539*.

Ioannidis, J. 2005. Why most published research findings are false. *PLoS Med.*, 2(8): 696–701.

Johnson, D. S. 1999. A theoretician's guide to the experimental analysis of algorithms. In *Data Structures, Near Neighbor Searches, and Methodology*.

Kool, W.; van Hoof, H.; Gromicho, J.; and Welling, M. 2021. Deep Policy Dynamic Programming for Vehicle Routing Problems. arXiv:2102.11756.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Attention, Learn to Solve Routing Problems! arXiv:1803.08475.

Lysgaard, J.; Letchford, A. N.; and Eglese, R. W. 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Program.*, 100(2): 423–445.

Máximo, V. R.; and Nascimento, M. C. 2021. A hybrid adaptive iterated local search with diversification control to the capacitated vehicle routing problem. *Eur. J. Oper. Res.*, 294(3): 1108–1119.

Nazari, M.; Oroojlooy, A.; Snyder, L. V.; and Takáč, M. 2018. Reinforcement Learning for Solving the Vehicle Routing Problem. arXiv:1802.04240.

Pessoa, A.; Sadykov, R.; Uchoa, E.; and Vanderbeck, F. 2020. A generic exact solver for vehicle routing and related problems. *Math. Program.*, 183(1): 483–523.

Uchoa, E.; Pecin, D.; Pessoa, A.; Poggi, M.; Vidal, T.; and Subramanian, A. 2017. New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur. J. Oper. Res.*, 257(3): 845–858.

Vidal, T. 2021. Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood. arXiv:2012.10384.

Vidal, T.; Laporte, G.; and Matl, P. 2020. A concise guide to existing and emerging vehicle routing problem variants. *Eur. J. Oper. Res.*, 286: 401–416.

Xin, L.; Song, W.; Cao, Z.; and Zhang, J. 2021. NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman Problem. arXiv:2110.07983.