Machine Learning for Evolutionary Computation - the Vehicle Routing Problems Competition

Weiyao Meng
School of Computer Science
University of Nottingham
Nottingham, UK
Weiyao.Meng2@nottingham.ac.uk

Rong Qu
School of Computer Science
University of Nottingham
Nottingham, UK
Rong.Qu@nottingham.ac.uk

Nelishia Pillay
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Nelishia.Pillay@up.ac.za

ABSTRACT

The Competition of Machine Learning for Evolutionary Computation for Solving Vehicle Routing Problems (ML4VRP) seeks to bring together machine learning and evolutionary computation communities to propose innovative techniques for vehicle routing problems (VRPs), aiming to advance machine learning-assisted evolutionary computation that works well across different instances of the VRPs. This paper overviews the key information of the competition.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; *Search methodologies*; • **Applied computing → Operations research**.

KEYWORDS

Machine learning, Evolutionary computation, Meta-heuristics, Vehicle routing

ACM Reference Format:
Weiyao Meng, Rong Qu, and Nelishia Pillay. 2024. Machine Learning for Evolutionary Computation - the Vehicle Routing Problems Competition. In *Proceedings of The Genetic and Evolutionary Computation Conference 2024 (GECCO '24).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3638530.3664046

1 INTRODUCTION

The Competition of Machine Learning for Evolutionary Computation for Solving the Vehicle Routing Problems (ML4VRP) [1] aims to serve as a vehicle to bring together the latest developments of machine learning-assisted evolutionary computation for VRPs. Results of current relevant research contain a lot of rich knowledge in evolutionary computation which is, however, often discarded or not further investigated. These include different features of the problem/solutions to inform or drive the evolution/optimisation, different settings/operators/heuristics in effective evolutionary algorithms, and observations/evaluations of the search/fitness space. These can all be retained and processed as data, serving as an excellent new problem domain for the machine learning community to enhance evolutionary computation [4].

VRP variants of different difficulties provide an ideal testbed to enable a performance comparison of machine learning-assisted computational optimisation. Fostering, reusing, and interpreting the rich knowledge from ML4VRP presents a challenge for researchers across disciplines, however, is highly rewarding to further advance human-designed evolutionary computation.

The previous ML4VRP competition at GECCO'23 focused on VRP with capacity and time window constraints (i.e., CVRPTW). Participants must develop machine learning techniques which can design and enhance evolutionary computational algorithms or meta-heuristics for solving CVRPTW. Submissions featured a range of innovative techniques, such as Graph Neural Networks (GNNs) [9], Graph Q-learning [10], and deep reinforcement learning. These approaches were applied to various aspects of the optimisation process, including parameter tuning, solution initialisation, offspring quality prediction, and operator selection.

Following the success of the previous competition, we are launching the competition at the GECCO'24, proposing two tracks in VRP, i.e., CVRP and CVRPTW. Participants are required to submit descriptions of the developed algorithms and the solutions for the provided CVRP/CVRPTW instances. The submissions will be evaluated on randomly selected instances (from the provided instances) using an evaluator available in the GitHub repository[2] dedicated to this competition. The most widely adapted evaluation function, i.e. to minimise the number of vehicles and total travel distance, is used to determine the best machine learning-assisted evolutionary algorithms for solving VRPs. The algorithms which produced the best average solution quality will receive the highest score.

2 PROBLEM INSTANCES AND FORMATS

2.1 The CVRP Track

The classical instances of Uchoa et al. (2017) [7] is one of the most widely studied CVRP benchmark datasets. This dataset covers different instance features, e.g., depot positioning, customer positioning, demand distribution, allowing a comprehensive evaluation of algorithm performance.

The problem instances provided in the competition are the Uchoa et al. (2017) instances with customers ranging from 100 to 400, covering different instance types.

[1]https://sites.google.com/view/ml4vrp

[2]https://github.com/ML4VRP/ML4VRP2024

GECCO '24, July 14–18, 2024, Melbourne, Australia

ACM ISBN 979-8-4007-0495-6/24/07
https://doi.org/10.1145/3638530.3664046

2.2 The CVRPTW Track

Solomon [6] and Hombeger-Gehring [3] instances are extensively utilised in the VRPTW literature. These datasets encompass six types of instances (C1, C2, R1, R2, RC1, and RC2), varying in customer locations, vehicle capacity, and time window constraints.

The problem instances in the competition are taken from the Solomon [6] dataset of 100 customers, and the Homberger and Gehring [3] dataset of 200 customers and 400 customers. The provided instances are randomly chosen from these three sized problems, covering different instance types.

2.3 Instance and Solution Formats

In this competition, we follow the convention used by the recent DIMACS VRP challenge[3] for instance and solution formats. CVRP instances are given in TSPLIB95 format [5], while CVRPTW instances follow the widely accepted standard format for this variant. Solutions for both tracks should adhere to the CVRPLIB format.

3 SOLUTION EVALUATION

Several studies on VRPs use the hierarchical objective function, prioritising the minimisation of the number of vehicles (routes) followed by total travel distance [1]. Some search algorithms in the literature adopt the weighted sum objective function [2]. This competition considers the dual objectives of minimising the number of vehicles (NV) and minimising the total travel distance (TD), as shown in Equation (1), where c is set to 1000 empirically [8].

$$c \times NV + TD \tag{1}$$

The competition's solution evaluator, implemented in Python, offers user-friendly access via the command line interface. It begins by assessing solution feasibility. If feasible, it computes and display the objective function value, the number of routes and total travel distance. If the solution is infeasible, the evaluator returns a failure. Full details can be seen in the competition's GitHub repository.

4 PARTICIPATION

Participation is open to individuals or teams. Participants should submit the following before 13 June 2024, and include team name, algorithm details, team leader, and primary affiliation:

- A short description of 1) the machine learning (e.g. supervised or unsupervised learning, reinforcement learning, deep learning, etc.) which designs, assists and enhances evolutionary algorithms; 2) the resulting algorithms (e.g. metaheuristics, evolutionary algorithms, etc.) supported by the machine learning for solving the CVRP/CVRPTW. No source code for the algorithm is required for this competition.
- The solutions in the specified format and the corresponding VRP instances, to be verified by the solution evaluator.

Participants may submit a two-page abstract by 8 April 2024, to be included in the GECCO Companion Proceedings if accepted. Participants are also invited to submit a full paper to a special issue on ML4VRP in a journal. Details will be made available on the competition website once the dates are finalised. We also encourage participants to attend GECCO'24.

[3]http://dimacs.rutgers.edu/programs/challenge/vrp/

5 SCORING SCHEME

The competition will evaluate the submitted solutions for a subset of the provided VRP instances, which will remain unknown to the participants until the results are released. To determine the winner and assess the performance of the competing machine learning-assisted algorithms, we will adopt a scoring scheme used in some of the competitions, which is based on Formula 1.

Formula 1 adopted a scoring scheme before 2010, where the top eight drivers in each race earned points: 10, 8, 6, 5, 4, 3, 2, and 1. The driver accumulating the most points across all races was declared the winner. This is adapted for this competition as follows.

Assume that there are m instances and n competing algorithms. For each instance, an ordinal value x is given representing the rank of the algorithm compared to the others ($1 \leq x \leq n$). The top eight ranking algorithms per instance will receive points: 10, 8, 6, 5, 4, 3, 2 and 1 (like Formula 1), while the rest receive no points for that instance. The points will be added across the m instances for each algorithm. The winner will be the algorithm with the highest total points. Therefore, if there are, for example, five instances in the evaluation, the maximum possible score is 50 points.

To break the ties when multiple algorithms obtain the same objective function value (rounded to 3 decimal places) on a given instance, the points awarded to the corresponding ranking positions are added together and then evenly distributed among them. This ensures the total number of points awarded for each instance remains the same, ensuring fairness in tie situations.

The winner of each track is the algorithm with the most points for that specific track. In the case of a tie in points, the algorithm with more wins is ranked first. If still tied, the winner goes to the algorithm which is ranked the most second places, and so on.

REFERENCES

[1] Olli Bräysy and Michel Gendreau. 2005. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science* 39, 1 (2005), 104–118.
[2] Olli Bräysy and Michel Gendreau. 2005. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation science* 39, 1 (2005), 119–139.
[3] Jörg Homberger and Hermann Gehring. 1999. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research* 37, 3 (1999), 297–318.
[4] Nelishia Pillay and Rong Qu. 2021. *Automated Design of Machine Learning and Search Algorithms.* Springer.
[5] Gerhard Reinelt. 1995. Tsplib95. *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg* 338 (1995), 1–16.
[6] Marius M Solomon. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* 35, 2 (1987), 254–265.
[7] Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. 2017. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research* 257, 3 (2017), 845–858.
[8] James D Walker, Gabriela Ochoa, Michel Gendreau, and Edmund K Burke. 2012. Vehicle routing and adaptive iterated local search within the hyflex hyper-heuristic framework. In *Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers.* Springer, 265–276.
[9] Xiangyu Wang and Yaochu Jin. 2023. An Ant Colony Algorithm Assisted by Graph Neural Networks for Solving Vehicle Routing Problems. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation.* 5–6.
[10] Peng Yue, Shiqing Liu, and Yaochu Jin. 2023. Graph Q-learning Assisted Ant Colony Optimization for Vehicle Routing Problems with Time Windows. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation.* 7–8.