Hyperparameter search in periodic vehicle routing problem

Ekaterina Grakova[1] , Martin Golasowski[1], Roberto Montemanni[2], Kateřina Slaninová[1] , Jan Martinovič[1], Jafar Jamal[2], Kateřina Janurová[1] and Matteo Salani[2]

[1] *IT4Innovations National Supercomputing Center, VŠB - Technical University of Ostrava, 17. listopadu 15/2172, 708 33 Ostrava, Czech Republic*
[2] *Dalle Molle Institute for Artificial Intelligence (IDISA, USI/SUPSI), Galleria 2, 6928 Manno, Switzerland*

Abstract. The large number of real-world applications have shown that the use of computational method for distribution process planning produces substantial savings. Many of these applications lead to problem generally known as Vehicle Routing Problem. The real-world applications are highly computationally demanding for larger instances. This article aims to show the possibilities and benefits of using hyperparameter search for solving the Periodic Vehicle Routing Problem for exhausted oil collection by execution on the supercomputing infrastructure using HyperLoom platform. HyperLoom is an open source platform for defining and executing scientific pipelines in a distributed environment. This experiment was run on the supercomputer Salomon operated by IT4Innovations.

1 Introduction

The logistic industry heavily relies on optimal route planning as it represents a critical task. Indeed, transportation costs account for up to 20% of the overall logistic cost [1]. The Vehicle Routing Problem (VRP) was proposed more than 50 years ago [2] and it is a challenging combinatorial optimization problem. Variants of the VRP have been proposed to respond to the variety of operational constraints arising in the distribution industry.

One of such variants is the Periodic Vehicle Routing Problem (PVRP) in which routes are constructed over a period of time (e.g. days). The problem has been introduced by Christofides and Beasley [3] and has been solved mostly through heuristic approaches [4 - 6], [16].

In [7] we considered a variant of the PVRP in which exhausted oil must be collected from customers. Customers can be visited more than once during the planning period and the number of times they are visited is also a decision variable of the problem, furthermore all vehicles must return to the depot each day as wasted oil must be properly disposed. In the considered problem, exhausted oil accumulates at customers' location at a fixed ratio expressed in liters per week. From this perspective our problem shares features with that of waste collection problem [8 - 10].

High performance computing (HPC) architecture can be a solution to solve such large tasks, and to obtain the optimal solutions. The supercomputing infrastructure allows us to solve the complicated optimization problems and to use the optimization algorithms to find the optimal solutions. Many of these algorithms depend on configuration settings that are typically hand-tuned in the course of evaluating the algorithm for a particular data set. Such parameter tuning (hyperparameter search) is often presented as being incidental to the algorithm and correctly setting these parameters improves the quality of the algorithm results [11, 17].

HyperLoom has been successfully used at IT4Innovations for distributed hyperparameter search within machine learning applications for pharma industry [18]. This application aims to be another use case proving its usability beyond the pharmaceutic domain. Scientific pipelines such those in machine learning compose of multiple data processing tasks. This article aims to show the possibilities and benefits of using hyperparameter search for solving the Periodic Vehicle Routing Problem for exhausted oil collection by execution on the supercomputing infrastructure using HyperLoom platform. For the experiments, we created extended test case for exhausted oil collection. Experiments were run on the supercomputer Salomon.

2 Periodic VRP for oil collection

We formally describe the PVRP for exhausted oil collection. We are given a set N of customers and a central depot, denoted as node 0. The set $V = N \cup 0$ forms the node set of a graph G whereas a set of arcs A is linking every pair of nodes in N. For each arc $a_{i,j}, (i,j) \in N \times N$ let $t_{i,j}$ and $d_{i,j}$ the travelling time and travelling distance between nodes i and j, respectively. Let p_j be the expected exhausted oil accumulation ratio expressed in liters per week. We are also given a set K of identical vehicles of capacity Q located at the central depot 0 and a set M of planning periods. The problem asks to define up to $|K|$ routes per period, one for each vehicle, possibly returning to the depot more than once per period and visiting a subset of customers. The planning problem has two objectives: minimize the total travelled distance and balance the workload between the single routes. This last objective is considered computing the standard deviation in routes total travelling time. Routes must respect vehicle's capacity constraints and customers should not accumulate more exhausted oil than that transportable by a single vehicle in a single visit.

2.1 Test case for exhausted oil collection

In a collaboration with Caritas Suisse, we were asked to apply our heuristic algorithms to an exhausted oil collection problem in the area of Bali, Indonesia. We have been given a set of 308 different locations for the collection of the exhausted oil. For each location we know the expected amount of oil that accumulates within one week. We were asked to recommend one out of three potential locations for the exhausted oil processing plant. The travel times between any two locations are known. To collect the exhausted oil, cans of 25 liters capacity each are used. Every collection tour starts at the depot with a vehicle filled with empty cans. Empty cans are then exchanged at customer location with full ones. Finally, cans are delivered at the processing plant. As an additional requirement, the travel time for each route is not allowed to exceed a given bound on working hours corresponding to eight hours in our tests.

In order to produce fairly balanced routes, we introduce a weight parameter w which is used to balance total travelled distance versus the standard deviation of route duration. The parameter ranges from 0 to 1. When w is set to 0, total travel time is the only considered objective, when w is set to 1, the standard deviation of route durations is the only considered objective.

More formally, a solution S to the problem is a set of routes $r_{k,p}, k \in \{1,\dots,|K|\}$, $p \in \{1,\dots,10\}$, where k is the vehicle index and p is the period index, starting at the depot, finishing at the depot, visiting a customer at most once.

Let $t(r_{k,p})$ represent the total travel time of route $r_{k,p}$ and $q_{v,j}$ represent the demand of customer $v \in V \backslash \{0\}$ accumulated until the customer gets visited at working day j. Let $s_v \in \{1,2,4\}$ be the decision variable representing the number of time customer v is visited in the planning period.

$$t(r_{k,p}) \le 8 \;\forall\, k \in \{1,\dots,|K|\}, p \in \{1,\dots,10\}, \tag{1}$$

$$\textstyle\sum_{v \in r_{k,p}} \lfloor q_{v,p} / c \rfloor \le Q \;\forall\, k \in \{1,\dots,|K|\},\; p \in \{1,\dots,10\}, \tag{2}$$

$$|\, r_{k,p} : k \in \{1,\dots,|K|\},\; p \in \{1,\dots,10\},\; v \in r_{k,p}\}| = s_v, \tag{3}$$

$$\left|\, r_{k,p} : k \in \{1,\dots,|K|\}, \quad v \in r_{k,p}\right\}| \le 1 \;\forall\, v \in V\{0\}, \\ p \in \{1,\dots,10\}, \tag{4}$$

Constraint (1) states that the total travel time for each route does not exceed the working time limit of eight hours and constraint (2) states that vehicle capacity is respected, here c is the amount of oil contained in one can and Q is expressed in number of cans. The third constraint relates the number of visits in the solution with variable s. Constraint (4) ensures that a customer is not visited twice per day. The objective is a combination of two objectives:

$$z_1(r) = \textstyle\sum_{k \in \{1,\dots,|K|\}} \sum_{p \in \{1,\dots,10\}} t(r_{k,p}), \tag{5}$$

$$z_2 = \sqrt{\frac{1}{10|K|} \textstyle\sum_{k \epsilon \{1\dots|K|\}} \sum_{p \epsilon \{1\dots10\}} t \left(\frac{(r_{k,p} - z_1(r))}{10|K|}\right)^2}\,, \tag{6}$$

The objective function can then be written as

$$z(r) = w\, z_1(r) + z_2(r) \tag{7}$$

We adopt a two level approach similar to the first assigned strategy used for PVRP [14]. The assignment of customers to vehicles and working days is managed on the first level of the heuristic and explored using local search. The second level optimizes the single routes of the different vehicles on different days. Nicely, the changes imposed by the first level local search algorithm affects a portion of the solution (two routes). Therefore, only two routes must be re-optimized and usually the starting solution is already of a good quality.

The heuristic starts with a random assignment of locations to vehicles and working days. It does so that the number of times a customer is visited in the planning period is compatible with its expected oil accumulation: none of the visit should occur after the customer accumulated more oil than the vehicle's capacity. For each vehicle and each working day an optimal route is computed. The obtained solution is the starting point for the local search algorithm. The local search exploits a shift operator and a swap operator. The shift operator takes a location assigned to a specific vehicle and a specific working day and shifts it to another vehicle and/or another working day. The swap operator considers two different locations and interchanges their assignments to vehicles and working days. In each iteration, the local search algorithm explores all the solutions in the local search neighborhoods induced by those two operators in a random order.

The second level optimization mechanism computes an optimal route by complete enumeration in the case where at most a given number of locations are assigned to a vehicle, otherwise a fast insertion heuristic approach is used.

If an improving solution is found, this solution replaces the current solution and a new iteration is started immediately. If there is no improving solution in the neighborhood of the current solution, the local search algorithm terminates, and the local optimal solution is returned by our approach as the final solution. Further details can be found in [7] and [15].

Histogram (Figure 1) reports the frequency of required running time for the algorithm to converge. Out of 75 runs, we observe that the algorithm convergence is almost always below 3 seconds. The proposed approach works very well for the given application anyway for the sake of challenging the HyperLoom platform a new and extended test case is necessary.

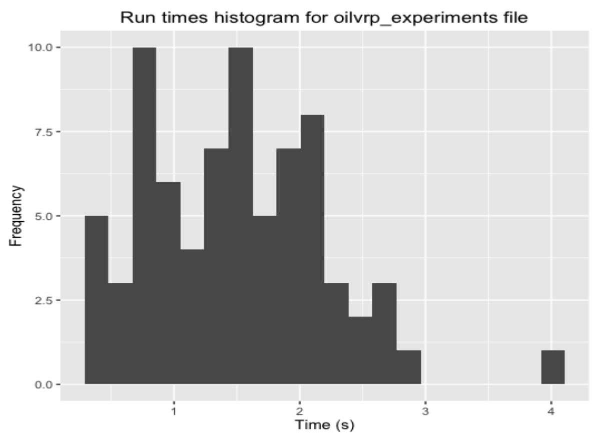

Figure 1. Run times histogram for test case

2.2 Extended test case for exhausted oil collection

We prepared an additional set of 1008 different locations for the oil collection problem. For each location we know the amount of oil. Every customer had a specific GPS location in Ostrava (Figure 2). We know the travel time between any two of the locations. Also, metric distances between the locations were created in Ostrava. We know the location of the central depot. The vehicles start at the central depot.

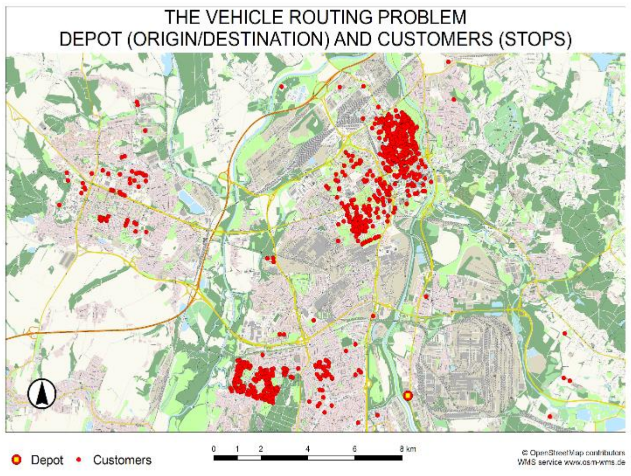
THE VEHICLE ROUTING PROBLEM
DEPOT (ORIGIN/DESTINATION) AND CUSTOMERS (STOPS)
Depot
Customers
0 1 2 4 6 8 km
© OpenStreetMap contributors
WMS service www.osm-wms.de

Figure 2. GPS locations in Ostrava

Planning time for algorithm testing was 10 working days.

3 HyperLoom for periodic VRP

HyperLoom is an open source platform for defining and executing scientific pipelines in a distributed environment developed at IT4Innovations. HyperLoom is able to efficiently execute millions of interconnected tasks on thousands of computational cores. In [12], the authors described a virtualization of HyperLoom – originally an HPC solution for defining and executing scientific workflows and compared the performance of the virtualized HyperLoom to the performance of the bare metal deployment.

We used HyperLoom to define and execute VRP hyperparameters sweep pipeline. HyperLoom enables to take an advantage of distributed systems by running multiple instances of VRP in parallel which promises an efficient usage of the provided resources. The quality of results for the heuristic algorithm depends on the adjustment of the configuration parameters of the algorithm. In this article, the following configuration parameters were used:

Table 1. Configuration parameters

Parameters name	**Configuration values**
Number_of_vehicles	15, 16, 17, 18, 19, 20
Depot_number	0
Weight	0.5, 0.05, 0.95
Random_Seed	1, 2, 3, 4
Horizon_days	10
Vehicles_capacity	34, 35, 36, 37, 38
Location_per_day	100, 101, 102, 103, 104, 105
Liters_per_day	200, 250, 300, 350

6000 combinations of configuration parameters were created for the algorithm. The experiments were performed on Salomon supercomputer, where each node has two – twelve – core Intel Xeon processors and 128 GB RAM (2xIntel Xeon E5 – 2680 v3, 2.5 GHz, 12 cores). We have carried out all the experiments on a 6, 12, 24, 48 identical physical computational nodes (Table 2). We have observed the dependence of the total execution time values on the number of nodes.

Table 2. Experiments results

Number of nodes	**Total execution time, [s]**	**Strong scaling efficiency, [%]**
6	5,128	98.52
12	2,841	88.92
24	1,443	87.53
48	839	75.27

Strong scaling efficiency was calculated for each experiment:

$$S = \frac{t_1}{(Nt_N)}\ 100\%, \qquad (8)$$

where:

t_1 - time to complete a work unit with 1 processing element, N- number of processing elements, t_N- time to complete the same unit of work with N processing elements.

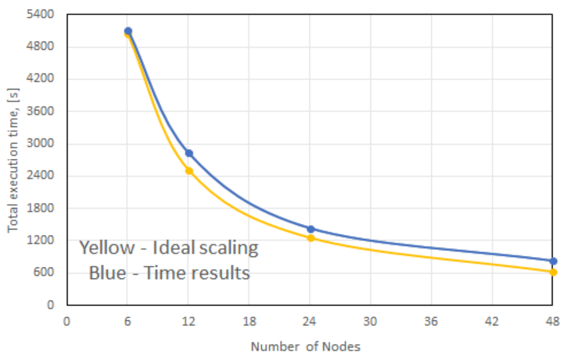

Figure 3. Strong Scaling

4 Algorithm results

The only result that was monitored during the experiments was the solution time of the algorithm which was between 62.999s and 201.839s for different configurations of input parameters described in Table 1. The histogram of solution time is given in Figure 4.

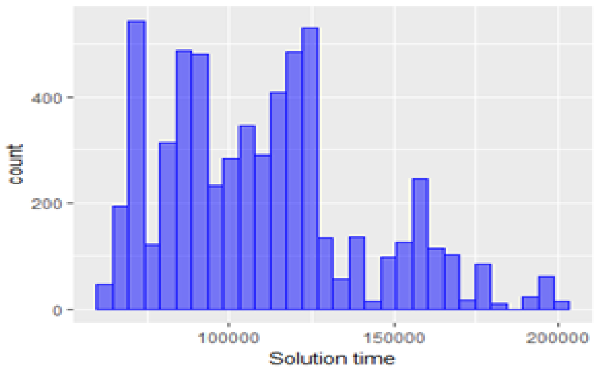

Figure 4. Histogram of solution time

We also assessed the dependence of solution time on the input parameter values and the importance of parameter values for total solution time.

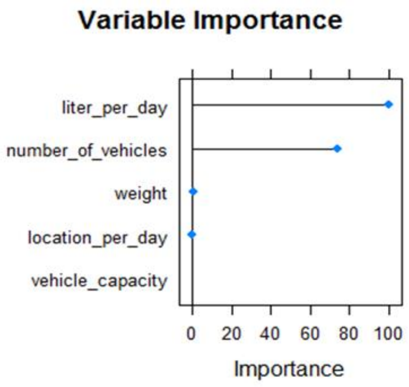

Figure 5. Parameters importance

The importance of the input parameters was determined as the proportion of the variance of the solution time that can be explained by the given input parameter (using the local regression model LOESS) [13]. This proportion was then recalculated as the relative measure where the importance of the input parameters is relative to the input parameter with the highest degree of importance (100%). The relative parameter importance is given in Figure 5 and together with correlation coefficients in Table 3.

As can be seen from the results, the total solution time is dependent only on two input parameters: liter_per_day and number_of_vehicles.

Table 3. Parameter importance and correlation coefficient

Parameter	**Parameter Importance**	**Correlation Coefficient**
liter_per_day	100%	-0.699
number_of_vehicles	74%	0.601
weight	0%	0.038
location_per_day	0%	0.002
vehicle_capacity	0%	0.000

The negative correlation coefficient for the liter_per_day input parameter means the fewer liter_per_day the higher solution time contrary to the number_of_vehicles, where the higher number_of_vehicles means the higher solution time.

5 Conclusion

In this article, we showed possibilities and benefits of using hyperparameter search for solving the Periodic Vehicle Routing Problem for the exhausted oil collection. HyperLoom platform was used for the definition of our hyperparameter search pipeline. We created an extended test case because the original instance of the problem was too small for test on the HPC infrastructures.

We have observed the dependence of the total execution time values on the number of nodes. For heuristic algorithm, we also assessed the dependence of solution time on the input parameter values and the importance of parameter values for total solution time. Using HyperLoom platform for our problem we showed that it is not necessary to parallelize used optimisation algorithm for Periodic Vehicle Routing Problem to use larger instances.

6 Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project 'IT4Innovations excellence in science - LQ1602', by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project 'IT4Innovations National Supercomputing Center – LM2015070' and by the ExCAPE project - the European Union's Horizon 2020 research and innovation programme under grant agreement No. 671555.

References

1. P., Toth, D., Vigo. The vehicle routing problem. SIAM monographs on discrete mathematics and

applications, *Society for Industrial and Applied Mathematics*, (2002)
2. G,B., Dantzig, J.H., Ramser. The truck dispatching problem. *Management Science,* **6**, 80–9, (1959)
3. N., Christofides, J.E., Beasley. The period routing problem. *Networks*, **14**, 2, 237–256, (1984)
4. A., Angelelli, M.G., Speranza. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, **137**, 2, 233–247, (2002)
5. J.F., Cordeau, M., Gendreau, G., Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems, *Networks*, **30**, 2, 105–119, (1997)
6. M., Gaudioso, G., Paletta. A heuristic for the periodic vehicle routing problem. *Transportation Science*, **26**, 2, 86–92, (1992)
7. D., Weyland, M., Salani, R., Montemanni, L.M., Gambardella. Vehicle routing for exhausted oil collection *Journal of Traffic and Logistics Engineering,* **1**, 1, 5–8, (2013)
8. B-I., Kim, S., Kim, S., Sahoo. Waste collection vehicle routing problem with time windows, *Computers & Operations Research*. **33**, 12, 3624-3642, (2006)
9. A.M., Benjamin, J.E., Beasley. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities, *Computers & Operations Research*, **37**, 12, 2270-2280, (2010)
10. K., Buhrkal, A., Larsen, S., Ropke. The waste collection vehicle routing problem with time windows in a city logistics context, *Procedia - Social and Behavioral Sciences*, **39**, 241-254, (2012)
11. J., Bergstra, D., Yamins, D.D, Cox. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. *In: Proceedings of the 30th International Conference on International Conference on Machine Learning*, **28**, 115-123, (2013)
12. M., Kuhn. Caret package, *Journal of Statistical Software*, **28**, 5, (2008)
13. V., Cima, S., Bohm, J., Martinovič, J., Dvorský, J.T., Ashby, V., Chupakhin. HyperLoom Possibilities for Executing Scientific Workflows on the Cloud. *CISIS Complex, Intelligent and Software Intensive Systems*, 397-406, (2017)
14. S., Baptista, R.C., Oliveira, E., Zuquete. A period vehicle routing case study. *European Journal of Operational Research*, **139**, 2, 220–229, (2002)
15. Stochastic Vehicle Routing - From Theory to Practice, Dennis Weyland. *University of Lugano (Lugano, Switzerland)* (2013)
16. E., Grakova, K., Slaninová, J., Martinovič, J., Křenek, J., Hazelka. Waste collection vehicle routing problem on HPC Infrastructure. *CISIM Computer Information Systems and Industrial Management Applications*. (2018)
17. A., Fred, et al. (2016). Pattern Recognition: Applications and Methods: 4th International Conference, ICPRAM 2015, Lisbon, Portugal, January 10-12, 2015, Revised Selected Papers. Springer. Retrieved March 15, (2018)
18. V., Cima, S., Böhm, J., Martinovič, J., Dvorský J., Janurová, T., Vander Aa, J.T., Ashby V., Chupakhin. HyperLoom: A platform for defining and executing scientific pipelines in distributed environments. *In: Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM '18). ACM, New York, NY, USA*, 1-6. (2018)