12th DIMACS Implementation Challenge: CVRP track

December 1st, 2021

The DIMACS Implementation Challenges address questions of determining realistic method performance where worst case analysis is overly pessimistic and probabilistic models are too unrealistic: experimentation can provide guides to realistic method performance where analysis fails. Experimentation also brings methodic questions closer to the original problems that motivated theoretical work. It also tests many assumptions about implementation methods and data structures. It provides an opportunity to develop and test problem instances, instance generators, and other methods of testing and comparing performance of methods. And it is a step in technology transfer by providing leading edge implementations of methods for others to adapt.

The 12th Implementation Challenge is dedicated to the study of Vehicle Routing problems, bringing together research in both theory and practice. This rendition of the Challenge is part of the DIMACS Special Focus on Bridging Continuous and Discrete Optimization and will be capped by a workshop hosted by DIMACS at Rutgers University in April 6-8, 2022. This Challenge is held in honor of David S. Johnson and includes activities dedicated to him and his many contributions to the study of methods.

1 Introduction

Among VRP variants, the CVRP, introduced in Dantzig and Ramser (1959), is the most central and is the one from which many others derive. Input to the CVRP consists of n locations (a depot and a set of $n-1$ customers), an $n \times n$ symmetric matrix D specifying the distance (or some other cost) to travel between each pair of locations, a quantity q_i that specifies the demand for some resource by each customer i, and the maximum quantity, Q, of the resource that a vehicle can carry. A feasible solution to the CVRP consists of a set of routes that begin and end at the depot, such that each customer is visited on exactly one route and the total demand by the customers assigned to a route does not exceed the vehicle capacity Q. An optimal solution for CVRP is a feasible solution that minimizes the total combined distance of the routes. In the CVRP competition it will be assumed that there are no restrictions to the number of routes in a solution.

As for all tracks of the 12th DIMACS Implementation Challenge, it is expected that participants of the CVRP track contribute with results, articles and discussions for both exact and heuristic methods. Those potentially rich exchanges will first happen in a free format, as messages in a mail list, significant contributions and decisions being consolidated as posts in the DIMACS page. Then, there will be presentations in the workshop and, finally, submissions to journal special issues. However, this document is about the implementation competition in its narrow sense.

The CVRP competition is devised at assessing competing methods in regards to both running time and solution quality. In its Phase One, competitors should perform all required runs in their own machines and send the resulting output files to the organizers. The top five ranked competitors in Phase One (the finalists) will advance to Phase Two, having to install their codes in the identical machines provided by the organizers. Phase Two runs will be performed by the organizers. The results and ranking of Phase Two will be presented during the workshop. The first ranked competitor in Phase Two will be declared the winner.

2 Participation

Participation in the competition is open to any person or group. However, it is necessary to perform a registration, informing names and affiliations for each person in the group, choosing a Competitor ID and a Solver Name. It is also necessary to provide the specification and identification of the machines, up to three, where Phase One runs will be performed.

3 Instances

CVRP is a well-studied variant with many instances appearing in the literature. Recently, a large and diversified set of benchmark instances was proposed in Uchoa et al. (2017). The so-called X instances were generated by systematically varying attributes like depot positioning (central, eccentric or random), customer positioning (random, clustered, random-clustered), demand distribution (seven possibilities) and average route size (five distinct ranges). The X instance set contains 100 instances, having from 100 to 1000 customers. These instances plus the following other instances from the literature will be used in the Phase One of the competition:

- E-n101-k8 and E-n101-k14, proposed in Christofides and Eilon (1969). Many authors assumed that the number of routes in a solution for those instances should be fixed to 8 and 14, respectively. However, in this competition it will be assumed that no such restriction exists for any instance. For example, solutions for E-n101-k8 having 9 routes will be accepted as feasible.

- CMT4 and CMT5, proposed in Christofides et al. (1979). The EUC_2D distances are not rounded in those instances.

- F-n135-k7, proposed in Fisher (1994).

- P-n101-k4, proposed in Augerat et al. (1995)

- tai385, proposed in Rochat and Taillard (1995). The EUC_2D distances are not rounded in that instance.

- Golden9 – Golden20, proposed in Golden et al. (1998), 12 instances having from 240 to 480 customers. The EUC_2D distances are not rounded in those instances.

- Antwerp1, Antwerp2, Brussels1, Brussels2, Flanders1, Flanders2, Ghent1, Ghent2, Leuven1, and Leuven2, 10 very large instances (having from 3,000 to 30,000 customers) proposed in Arnold et al. (2019). We note that some of those instances are so large that only storing the EUC_2D distances as a full matrix may already cause an out-of-memory error in a typical machine. Solvers should avoid doing that for those huge instances.

Phase One of the competition will also include 12 new instances, specially contributed to the Challenge. Those new instances are derived from real problems, their distance matrices are obtained from actual metropolitan street/road networks. Such distances are explicitly given in the instance files.

- Loggi-n401-k23, Loggi-n501-k24, Loggi-n601-k19, Loggi-n601-k42, Loggi-n901-k42, Loggi-n1001-k31, kindly contributed by Loggi, extracted from a large dataset of real vehicle routing and facility location instances[1]. Those six instances correspond to problems defined in three metropolitan areas in Brazil: Belém, Brasília and Rio de Janeiro. Actual road distances (in multiples of 10 meters) are provided. The distances are made symmetric by averaging the distance over both directions.

- ORTEC-n242-k12, ORTEC-n323-k21, ORTEC-n405-k18, ORTEC-n455-k41, ORTEC-n510-k23, ORTEC-n701-k64, kindly contributed by Wouter Kool (ORTEC). Those instances are derived from a real US based grocery delivery service. Distances correspond to real driving times (in seconds). The driving times are made symmetric by averaging the time over both directions.

All those 141 instances can be found in CVRPLIB[2]. In fact, that repository will be actively used to support the CVRP track of the competition.

Phase Two of the competition will use 100 new instances (unknown to the competitors before the results are presented) statistically similar to the X instances, obtained by the same random generator using a different seed.

[1] `https://github.com/loggi/loggibud`
[2] `http://vrp.galgos.inf.puc-rio.br/index.php/en/`

4 Scoring System

For each test instance, a competing solver will be evaluated according to the primal integral (Berthold, 2013) for the entire execution. Let BKS be the value of the best known solution for a given instance and define $v(0) = 1.1 \times BKS$. Let T be the maximum running time (in seconds) defined for a given instance and suppose that the solver finds a sequence of n solutions better than $v(0)$ and with decreasing value within that time limit. For each solution $i = 1, \ldots, n$, let $v(i)$ be its value and let $t(i)$ be the time (in seconds) it was found. Define $t(0) = 0$. The (normalized) primal integral is computed as:

$$PI = 100 \times \left(\frac{\sum_{i=1}^{n} v(i-1).(t(i) - t(i-1)) + v(n).(T - t(n))}{T \times BKS} - 1 \right).$$

Note that a solver that does not find any solution better than $v(0)$ (so $n = 0$) gets $PI = 10$, the worst possible evaluation. In principle, if the solver finds solutions better than BKS, it is possible to have negative values for PI.

The PI results of individual instances are aggregated into a single score using a points-based method: for each instance tested, points are awarded according to the scoring system used by Formula 1 between 2003 and 2009. For each instance, all competing solvers are ranked according to their individual PI value. The best solver gets 10 points, the second 8, then 6, 5, 4, 3, 2, 1. In case of ties (not very likely, since the times $t(i)$ will be measured with a precision of 3 decimal places, i.e., miliseconds), the points at play are evenly split among the solvers involved. For examples, if two solvers are tied in the first position, each solver will receive $(10+8)/2 = 9$ points; if three solvers are tied in seventh place, each solver will receive $(2 + 1 + 0)/3 = 1$ point. The total point score of a solver is then the sum of its points over all test instances. The competitor rankings will be based on total point score, ties being broken by the average PI over all test instances.

5 Computational Environment

The competing solvers should run in a single processor thread, under a Unix/Linux OS. The organizers of the challenge will provide a Controller executable code[3] that will run the competitor Solver. Every time Solver finds an improving solution, it should immediately write it to its standard output (make sure to also call a flush command for clearing the output buffer). The Controller will read each solution (through a Unix pipeline), check its feasibility and record the corresponding elapsed time. Controller will kill Solver process after the given time limit and compute the Primal Integral of the run. If Solver stops by itself (or crashes), Controller still computes a valid Primal Integral. For example, the command

```
%build/CVRPController Wolverine InstancesRounded/X-n120-k6.vrp 1 2367
```

[3] https://github.com/laser-ufpb/CVRPController

```
1800 13332 1 Solver1
```

calls "Solver1 X-n120-k6.vrp 1 1521" and produces an output file DIMACS-CVRP-Wolverine-X-n120-k6.out like:

```
12th DIMACS Implementation Challenge: Vehicle Routing
CVRP track
Controller version: November 17, 2021
Competitor: Wolverine
Ubuntu 18.04.5 LTS
Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
hostid: 8323329
PassMark Single Thread Benchmark: 2367
Time factor: 1.18 (baseline 2000)
Instance: X-n120-k6
Distance type: 1
Standardized Time limit: 1800 secs
Local Machine Time Limit: 1521 secs
Base solution: 14665.200
BKS: 13332
Optimal: 1
Wed 17 Nov 2021 03:35:32 PM CET
timestamp: 1637159732
Solution value, local machine time, standardized time
13934 0.007 0.008
13644 0.014 0.017
13577 0.144 0.170
13553 0.301 0.356
13546 0.325 0.385
13535 0.363 0.430
13461 0.420 0.497
13446 0.608 0.720
13431 0.763 0.903
13405 0.793 0.939
13349 1.044 1.236
13333 1.071 1.268
13332 1.124 1.330
Primal Integral: 0.0008938794
```

All times are **wall clock times**. It is up to the competitors (in Phase One) and to the organizers (in Phase Two) to perform the runs in a machine that is not heavily loaded.

The parameters of Controller are the following:

1. Competitor ID. Each competitor (a person or a group) should register an ID. As that id will be used in the name of the output file, it should only contain characters that are acceptable for that purpose in Unix (no accents or special characters).

2. The instance file in TSPLIB format.

3. Distance type: 0 = EUC_2D not rounded; 1 = EUC_2D rounded; 2 = Explicit.

4. The CPU mark. In order to compensate for different processor speeds, Controller will standardize (i.e., scale) times according to the CPU marks provided by PassMark Single Thread Performance[4]. Currently, the top CPU mark is 4,202, while mid-range desktop processors have marks around 2,000. So, we choose the mark 2,000 to define our standardized times. This means that if a run is performed in a processor Intel Core i9-9900T @ 2.10GHz that has mark 2,400, all local elapsed times will be multiplied by 1.2 to obtain the corresponding standardized times. This also means that a standardized time limit of 1,800 seconds will actually correspond to 1,500 seconds in that particular machine.

   - **Runs must be performed in a processor listed in PassMark and having a mark of at least 1,500.** In fact, the machine specifications (given by a sample of the Controller output) should be sent to the organizers at the registration. They will provide the marks to be used in the actual competition, based on the latest update of PassMark. Runs using a different mark will be disqualified.

5. The time limit: 1,800 seconds for instances with $n \leq 200$, 3,600 seconds for $200 < n \leq 400$, and 7,200 seconds for $n > 400$.

6. BKS. The value of the best known solution is used for calculating $v(0)$.

7. Optimal? If 1, means that the BKS is proven to be optimal. Controller saves computing resources by killing a Solver that already obtained an optimal solution. Of course, this does not affects the Primal Integral of the run.

8. Solver Name. It will called by Controller. The three parameters of the call are: instance file, distance type, and local machine time limit. The last parameter can be used by the solver in its strategy.

Some additional information:

- At the registration, competitors have to state that they already successfully tested Controller in each machine (up to three) that will be used for their Phase One runs. No later complaints will be possible. In fact, the competitors have to send to the organizers a sample Controller output for each machine. Even if a competitor plans to execute all runs in a single machine, we recommend registering at least a second machine as backup.

[4] `https://www.cpubenchmark.net/singleThread.html`

- Running all instances of Phase One in sequence would take about 8 days in a machine with baseline mark of 2,000. The instances will be divided into three groups and the organizers will provide three script generators for running the instances in each group. If desired, competitors may run each script in distinct registered machines. They may also run the scripts sequentially on the same machine or even in parallel on the same machine, if the machine has enough resources (cores and memory) to not slow down the runs. **In any case, each script has to be fully run on the same machine.** The organizers will check whether the timestamps in the output files of the instances in the same script are consistent. Competitors that not follow that rule will be disqualified.

  The call to script generators will be like:

  ```
  % sh genScript1.sh Wolverine 2367 Solver1 > CVRP-Script1.sh
  ```

- In Phase One, competitors should collect all Controller output files and send them to the organizers in a single zipped file until the scheduled deadline. After the Phase One ranking is published, all output files from all competitors will be made available in the DIMACS web page of the CVRP competition.

- Competitors should make sure that third-part software used by their solvers (like CPLEX, Gurobi or other MIP solvers) are parameterized for only using a single thread.

- **It is up to the five competitors qualified to Phase Two (the finalists) to install their solvers in the machines provided by the organizers.** In case of solvers that use third-part software, they should also install those software and (if needed) provide the proper licenses. Failure to do that until the scheduled deadline disqualifies the competitor.

- **The finalists are required to provide a document in article format describing the methods used in their solvers.** That document should be limited to 6 pages, not counting possible appendices with detailed tables of results. Failure to send that document until the scheduled deadline disqualifies the competitor.

- The finalists are automatically invited to make a presentation at the workshop (either physically or online) describing the methods used in their solvers.

- The finalists are encouraged to submit a full article to one the 12th DIMACS Challenge journal special issues. However, those articles will pass by the usual reviewing process. There is no guarantee that they will be accepted for publication.

6 Instance Format

Instances in CVRPLIB are in TSPLIB95 format [5]. Here is a sample instance with 5 costumers:

```
NAME : toy.vrp
COMMENT : Example of a CVRP instance with EUC_2D distances
TYPE : CVRP
DIMENSION : 6
EDGE_WEIGHT_TYPE : EUC_2D
CAPACITY : 30
NODE_COORD_SECTION
1 38 46
2 59 46
3 96 42
4 47 61
5 26 15
6 66 6
DEMAND_SECTION
1 0
2 16
3 18
4 1
5 13
6 8
DEPOT_SECTION
1
-1
EOF
```

All 129 instances from the literature listed for Phase One use EUC_2D distances (but CMT4, CMT5, tai385, and Golden9-20 instances do not follow the TSPLIB95 convention of rounding to the nearest integer). All the 100 instances that will be generated for Phase Two will use standard EUC_2D distances. In all those instances the depot is always in location 1.

The 12 new contributed instances use real distances obtained from street/road networks. All of them follow the TSPLIB95 LOWER_ROW convention, as in the following example:

```
NAME : toy2.vrp
COMMENT : Example of a CVRP instance with explicit distances
TYPE : CVRP
DIMENSION : 4
EDGE_WEIGHT_TYPE : EXPLICIT
EDGE_WEIGHT_FORMAT : LOWER_ROW
```

[5] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf

```
NODE_COORD_TYPE : TWOD_COORDS
CAPACITY : 2
EDGE_WEIGHT_SECTION
23
29 9
17 35 15
NODE_COORD_SECTION
1 0 0
2 0 20
3 14 18
4 17 0
DEMAND_SECTION
1 0
2 1
3 1
4 1
DEPOT_SECTION
1
-1
EOF
```

Coordinates are given for drawing purposes only, the distances are those explicitly given.

7 Solution Format

Solutions should be represented in the CVRPLIB format. For example, the optimal solution to toy.vrp in that format is:

```
Route #1: 1 4
Route #2: 3 2 5
Cost 265
```

For historical reasons, CVRPLIB solution format uses a convention that is a bit different from TSPLIB95: customers are numbered from 1 to $n-1$. So, that solution corresponds to routes $1-2-5-1$ and $1-4-3-6-1$ in TSPLIB95 numbering.

The optimal solution to toy2.vrp is:

```
Route #1: 3
Route #2: 2 1
Cost 95
```

Some remarks:

- Controller ignores all lines of Solver output that do not start with “Route” or “Cost”.

- The routes in a solution should be sequentially numbered.
- No empty routes are allowed.
- After reading a "Cost" line, Controller assumes that the solution is complete and check its feasibility. Unfeasible (or out-of-format) solutions are ignored. If the solution is feasible, Controller calculates its value. The actual solution value (not the number after "Cost") is considered. Solutions that are not better than $v(0)$ or do not improve upon the previous best solution are ignored.

8 CVRP Competition Schedule

The relevant dates for the CVRP competition are:

November 17th, 2020 – Start of the CVRP track competition. Controller ready to be downloaded and tested by potential competitors. Informal discussions on mail list, significant contributions and decisions being consolidated as posts in the web page.

November 11th, 2021 – Addition of 12 contributed instances to Phase One.

December 1st, 2021 – Release of the definitive version of this document, the competition rules.

December 8th, 2021 – Deadline for registration of competitors and their machines.

December 15th, 2021 – Official list of competitors posted. Registered competitors receive their CPU marks and running scripts.

January 16th, 2022 – Deadline for competitors to send all output files for Phase One (can only be done once).

January 23th, 2022 – Results of Phase One posted.

February 1st, 2022 – Deadline for the top five ranked competitors in Phase One (the finalists) to send the document in article format describing their method (it is recommended to start writing that document with sufficient advance).

February 7th, 2022 – Deadline for the finalists to install their codes in the machines indicated by organizers.

April 6-8th, 2022 – 12th DIMACS Challenge Workshop. Presentations by the finalists of their methods. Announcement of Phase Two results.

References

F. Arnold, M. Gendreau, and K. Sörensen. Efficiently solving very large-scale routing problems. *Computers & Operations Research*, 107:32–42, 2019.

P. Augerat, J. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M, Université Joseph Fourier, Grenoble, France, 1995.

T. Berthold. Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6):611–614, 2013.

N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, 20(3):309–318, 1969.

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, volume 1, pages 315–338. Wiley Interscience, 1979.

G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

M. Fisher. Optimal solution of vehicle routing problem using minimum k-trees. *Operations Research*, 42:626–642, 1994.

B. Golden, E. Wasil, J. Kelly, and I. Chao. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In *Fleet management and logistics*, pages 33–56. Springer, 1998.

Y. Rochat and É. D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1):147–167, 1995.

E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.