12th DIMACS Implementation Challenge: VRPTW 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

The Vehicle Routing Problem with Time Windows (VRPTW) is a more constrained variant of the CVRP in which each customer requires delivery within a specified interval of time called its "time window". Input to the VRPTW consists of n locations (a depot and a set of $n-1$ customers), an $n \times n$ symmetric matrix D specifying the both the distance and the time 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. In addition, each node i is specified with a length of time s_i denoting the time it takes to serve customer i and a time window $[t_i, T_i]$, where $t_i < T_i$, within which delivery must begin. A vehicle is allowed to arrive at a customer location before the beginning of the time window, but it must wait for the window to "open" to make the delivery. A delivery cannot be started after the time window closes. In this way, the Implementation Challenge considers the time-window constraints to be hard constraints. Finally, there is a limit V on the number of available vehicles. A feasible solution to VRPTW consists of a set of V or fewer routes that begin and end at the depot, such that each customer is visited on exactly one route within its specified time window and the total demand assigned to a route does not exceed the vehicle capacity Q. The objective is finding a feasible solution that minimizes the total combined distance of all routes.

As for all tracks of the 12th DIMACS Implementation Challenge, it is expected that participants of the VRPTW 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 VRPTW 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 three 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

For the Phase One of the competition, two classic sets of benchmarks instances will be used:

- The 56 instances proposed in Solomon (1987), each containing 100 customers;

- The 300 instances proposed in Homberger and Gehring (1999), having 200, 400, 600, 800, and 1,000 customers.

All those 356 instances can be downloaded in `https://github.com/laser-ufpb/VRPTWController/archive/refs/heads/master.zip`.

Solomon and Hombeger-Gehring instances are extensively used in the VRPTW literature. Yet, different conventions are found, making direct comparisons harder. Some authors use an hierarchical objective function, first minimizing the number of used vehicles and then minimizing the total route distance. However, the competition will adopt the convention (already used in several recent works, like Baldacci et al. (2011) and Pecin et al. (2017)) of only minimizing the total distance. So, it will be checked whether the solution respects the maximum number of vehicles available V, but there is no bonus for using less vehicles. Matrix D containing the distances and times is obtained from the location coordinates, by computing the Euclidean distances truncated to one decimal place. More formally, the distances (and times) d_{ij} are computed as follows:

$$d_{ij} = \frac{\lfloor 10 e_{ij} \rfloor}{10},$$

where e_{ij} is the Euclidean distance between locations i and j. This rounding convention is the one originally proposed in Solomon (1987).

Phase Two of the competition will use 300 new instances (unknown to the competitors before the results are presented), that will be obtained from the Hombeger-Gehring instances by only changing the coordinates of the depot. The new depot location will be random, but subject to the constraint that the resulting instance still has feasible solutions.

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[1] 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/VRPTWController Wolverine Instances/Solomon/R101.txt 2367
1800 1637.7 1 Solver1
```

calls "Solver1 R101.txt 1521" and produces an output file DIMACS-VRPTW-Wolverine-R101.out like:

```
12th DIMACS Implementation Challenge: Vehicle Routing
VRPTW 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: R101
Standardized Time limit: 1800 secs
Local Machine Time Limit: 1521 secs
Base solution: 1801.470
BKS: 1637.7
Optimal: 1
Wed 17 Nov 2021 10:30:34 PM CET
timestamp: 1637184634
```

[1]https://github.com/laser-ufpb/VRPTWController

```
Solution value, local machine time, standardized time
1719.3 0.068 0.080
1718.8 0.132 0.156
1710.1 0.148 0.175
1708.4 0.179 0.212
1706.9 0.187 0.221
1687.0 0.207 0.245
1686.7 0.218 0.258
1654.0 0.442 0.523
1648.4 0.455 0.538
1641.3 0.480 0.568
1640.9 0.549 0.650
1639.6 0.586 0.694
1638.8 0.658 0.779
1637.7 0.678 0.802
Primal Integral: 0.0013784305
```

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 the format defined in Section 6.

3. 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[2]. 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.

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

4. The time limit: 1,800 seconds for instances with $n \leq 201$, 3,600 seconds for $201 < n \leq 801$, and 7,200 seconds for $n > 801$.

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

6. 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.

7. Solver Name. It will called by Controller. The two parameters of the call are: instance file 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.

- Running all instances of Phase One in sequence would take about 15 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 > VRPTW-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 VRPTW 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 three 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 for VRPTW are given in what has become the *de facto* standard for this variant. Each instance is in a separate file. The files adhere to the following conventions:

```
[Instance name]
/* empty line */
VEHICLE
NUMBER     CAPACITY
  V           Q
/* empty line */
CUSTOMER
CUST NO.  XCOORD.   YCOORD.    DEMAND   READY TIME  DUE DATE   SERVICE TIME
/* empty line */
    0       x0        y1         q0         t0          T0            s0
    1       x1        y2         q1         t1          T1            s1
   ...     ...       ...        ...        ...         ...           ...
   10     x10       y10        q10        t10         T10           s10
```

Customer 0 is the depot. The input file for the toy.txt instance is shown below as an example.

```
toy

VEHICLE
NUMBER     CAPACITY
  3          50

CUSTOMER
CUST NO.  XCOORD.    YCOORD.    DEMAND   READY TIME   DUE DATE   SERVICE TIME

    0      50         50          0          0       200          0
    1      35         65         10         45       50           10
    2      30         55         30         50       80           10
    3      40         45         10         0        20           10
    4      55         70         20         0        100          10
    5      75         50         10         50       70           10
    6      54         35         20         17       20           10
```

7 Solution Format

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

```
Route #1: 3 1 2
Route #2: 6 5 4
Cost 153.5
```

That solution corresponds to routes $0 \to 3 \to 1 \to 2 \to 0$ and $0 \to 6 \to 5 \to 4 \to 0$. 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 VRPTW Competition Schedule

The relevant dates for the VRPTW competition are:

> August 23th, 2021 – Start of the VRPTW 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.

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 three 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

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011. doi: 10.1287/opre.1110.0975.

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

J. Homberger and H. Gehring. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 37(3):297–318, 1999. doi: 10.1080/03155986.1999.11732386.

D. Pecin, C. Contardo, G. Desaulniers, and E. Uchoa. New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29(3):489–502, 2017.

M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.