12th DIMACS Implementation Challenge: IRP track

November 21, 2021

1 Introduction

Among the CVRP variants, The Inventory Routing Problem (IRP) is one of the most challenging. As for all tracks of the 12th DIMACS Implementation Challenge, it is expected that participants of the IRP 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 algorithm evaluation portion of the competition in its narrow sense.

2 Inventory Routing Problem (IRP) Statement

The Inventory Routing Problem (IRP) is one of the most challenging of the VRP variants, as it combines both inventory management and routing decisions into a single problem.

There are many different variants of the IRP, but this Challenge will consider a single variant that we believe to be both representative of the problem class and difficult because it involves both inventory and routing decisions and costs.

The IRP consists of routing a single commodity from a single supplier (called the depot in the following) to multiple customers over a given planning horizon. Input for IRP consists of locations for the depot and the set of n customers $\{1, \ldots, n\}$, a time horizon consisting of T time periods $\{1, \ldots, T\}$, and a fleet of M vehicles $\{1, \ldots, M\}$. The depot is located at node 0. Each vehicle has capacity Q.

In each time period t, r_{0t} units of the commodity are made available at the depot, and r_{it} units are consumed by customer i. Each customer $i \in \{1, \ldots, n\}$ begins with an initial inventory level of I_{i0} and the depot begins with an initial inventory level of I_{00}. Each customer $i \in \{1, \ldots, n\}$ must maintain an inventory level of at least L_i and at most U_i and will incur a per-period cost of h_i for each unit of inventory held. There is no maximum inventory level defined for the depot, but it cannot fall below 0, and there is per-period inventory holding cost of h_0.

A matrix D specifies the distance (or some other cost) to travel between each pair of nodes.

The IRP is the problem of determining, for each time period t, the quantity to deliver to each customer i and the routes by which to serve those customers. An optimal solution is one that minimizes the total cost while assuring that constraints on vehicle capacity, vehicle routing, and node inventory levels are satisfied at all times. The total cost of a solution includes both the total of the inventory holding costs at all nodes (customers and depot) and the costs of the vehicle routes. A feasible solution must assure that:

- No customer stock-outs occur
- The depot has enough resource to meet deliveries in each period
- Each route begins and ends at the depot and honors the vehicle capacity constraint
- In each time period, a customer receives at most one delivery

This is the so-called Maximum Level (ML) replenishment policy, where any quantity can be delivered to the customers, provided that the maximum inventory level U_i is not exceeded and the minimum inventory level L_i is satisfied.

Maximum and minimum inventory level constraints: The maximum inventory level constraint establishes that, at each time period t, the sum of the inventory level in $t-1$ plus the quantity received in t should not exceed U_i. The minimum inventory level constraint establishes that the inventory level at time t should not be lower than L_i for each t.

2.1 Solution cost

The total cost of a solution is the sum of the total cost of travel and the total cost of inventory, at the customers and at the depot. The total travel cost is based on Euclidean distance (see below). It is the total distance traveled by all vehicles over all time periods. The inventory cost is the sum of the cost of inventory at each customer node over all time periods plus the cost of inventory at the depot over all time periods. For each node i, the inventory cost in a given time period is given by multiplying h_i by the inventory level at the time period. The inventory level is computed as described below.

Travel cost: All instances are Euclidean, so the node locations provided in the input file correspond to Euclidean coordinates for the nodes. To calculate the transportation cost, we will use a rounded Euclidean distance between consecutive customers i and j. That is calculated using the following formula to compute d_{ij}, the cost to travel from customer i to customer j.

$$d_{ij} = int((sqrt((x_i - x_j)^2 + (y_i - y_j)^2) + 0.5)$$

where “sqrt” is the square root and “int” is a function truncating to the integer value.

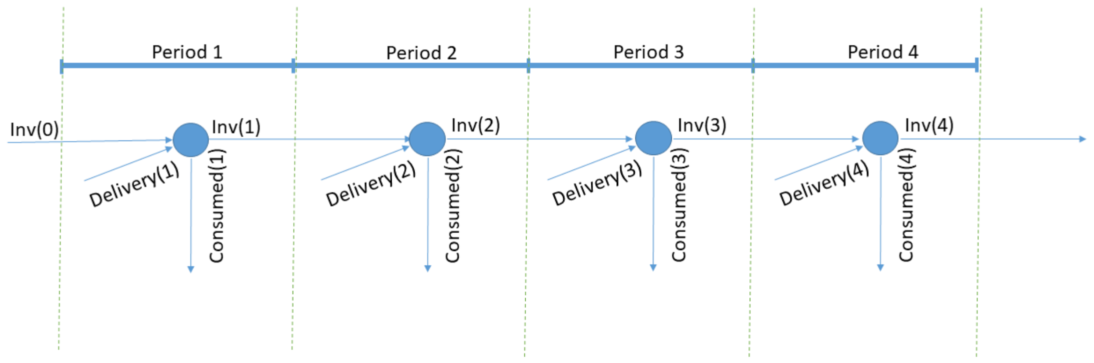
Inventory cost: For each customer i, the inventory in time period t is the inventory from time period $t-1$ plus the amount delivered in period t minus the amount consumed in period t. Letting q_{it} be the quantity consumed by customer i in period t, the inventory level is:

$$I_{it} = I_{it-1} + q_{it} - r_t$$

Figure 1 shows a simple flow of goods at a customer node over four periods.

For the depot, the inventory in time period t is the inventory from period $t-1$ plus the amount produced in time period t minus the total amount delivered to customers in period t.

Inventory costs are accounted for in all time periods $\{1, \ldots, T\}$. Note that inventory costs in time period 0 are not considered.



Figuer 1: This picture shows the flow balance of resources at a customer node over four time periods.

- Delivery(t) are the resources delivered in time period t.
- Consumed(t) are the resources consumed in time period t.
- Inv(t) is the inventory carried forward from time period t.

2.2 Inventory Bounds

Although Figure 1 implies that consumption and delivery occur simultaneously in each time period, that would not be the case in practice. In the IRP Challenge, we make the tacit assumption that delivery in period t occurs before consumption leading to the following constraints.

Lower Bounds: $I_{it} \geq L_i$

Upper Bounds: $I_{it-1} + q_{it} \leq U_i$

3 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.

4 Instances

IRP instances are provided in two zipped directories containing, respectively, small instances and large instances. Instances can be downloaded on the IRP track webpage.

All instances are Euclidean.

In all instances, production at the depot and consumption at the customers is constant over all time periods. In this way, the test instances are less general than the problem statement given in Section 2 and solvers may assume this fact.

Small instances have two possible values for the number of time periods, T, which will be either 3 or 6. T is 6 in all large instances.

Both the small and large instance sets are comprised of two subsets corresponding to instances with a low inventory cost and instances with a high inventory cost.

A description of the instances is provided in the PDF file available on the IRP track webpage.

Competing solvers will be evaluated for their performance on the set of small and large instances as provided in the website of the track.

5 Input File Format

Each IRP instance has the following format:

- The first line contains four parameters: the number of nodes including the depot $(n+1)$, number of time periods T, vehicle capacity Q, and number of vehicles M.
- The second line describes the depot and provides: depot identifier (which is "0"), x coordinate, y coordinate, starting inventory level I_{00}, daily production r_{0t}, inventory cost h_0.
- The n lines that follow each describes a customer and provides: customer identifier (i), x coordinate, y coordinate, starting inventory level I_{i0}, maximum inventory level U_i, minimum inventory level L_i, daily consumption r_{it}, inventory cost h_i.

6 Solution File Format

The solution file name must be of the form "out_INSTANCE.txt", where "INSTANCE" is the name of the instance. For example, the output file associated with the instance "Sabs1n20_1_H3" would be out_Sabs1n20_1_H3.txt.

The output file should be formatted such that the routes for each time period are provided together, followed by the routes for the next time period, and so on. For a given time period, the route for each vehicle $(1, \ldots, M)$ is provided on a separate line. Routes for each time period should be numbered from 1 to M. Since each route begins and ends at the depot, each route provided in the output should begin and end at node 0. The customers on the route should appear in order, each followed by the amount of demand delivered to them on this route in parentheses. The last six rows of the file summarize the solution cost and time by providing:

- Total transportation cost
- Total inventory cost at customer locations
- Total inventory cost at the depot
- Total cost (sum of the three terms above)
- Processor as listed in PassMark (see Section 8.1)
- Solution time (wall clock)

In the following, we will refer to time periods as days. The format of the output is as follows:

Day 1
Route 1: 0 – customer (quantity delivered) – . . . customer (quantity delivered) – 0
Route 2: 0 – customer (quantity delivered) – . . . customer (quantity delivered) – 0
⋮
Route M: 0 – customer (quantity delivered) – . . . customer(quantity delivered) – 0
Day 2
Route 1: 0 – customer (quantity delivered) – . . . customer (quantity delivered) – 0
⋮
Route M: 0 – customer (quantity delivered) – . . . customer (quantity delivered) – 0
⋮
Day T

Route 1: . . .
⋮
Route M: . . .
Total transportation cost
Total inventory cost at customer locations
Total inventory cost at the depot
Total cost of the solution
Processor
Solution time

A few points to clarify details about the format:

- Nodes in routes are separated by "blank_space – blank_space"
- The quantity delivered to each customer is reported after the customer node identifier this way: "customer_node blank_space left_parenthesis blank_space quantity blank_space right_parenthesis".
- The following is an example of example of a route that delivers 10 units to node 2 and 15 units to node 5:

  0 – 2 (10) – 5 (15) – 0
- It may be that not all vehicles make deliveries in every time period, but M routes must be output nonetheless. An empty route (not visiting any customers) is given this way:

  0 – 0

Output files will be read and processed automatically. Therefore, **files that do not comply with the format described above will be discarded**.

7 Solution Checker

We are grateful to Stephan Beyer for making a feasibility checker available to Challenge participants at this website: https://github.com/sbeyer/dimacs-irp-verifier.

The code to verify IRP solutions is written in Python, and instructions for its use are provided in the associated README file.

8 Algorithm evaluation

The Implementation Challenge will recognize solvers in a variety of ways. Some recognitions will be across all variants, while others will be within a single variant. The full VRP Implementation Challenge committee will review all entries with respect to cross-cutting recognition categories, such as novelty of the approach taken and simplicity and versatility of a solver to work well on multiple different variants. We hope to recognize multiple entries for these attributes. Additionally, each variant will hold its own competition in which entries are scored based on a set of quantitative measures to produce a ranking that does not depend on the subjective assessments of the committee. A set of top-scoring entries will be invited to present at the workshop, provided that they conform to the competition rules and document their approach and results in a suitable extended abstract.

This section describes the quantitative algorithm evaluation for the IRP portion of the Challenge. Stated briefly, a solver will be assessed based on the quality of the solutions obtained within a given amount of time. To mitigate the effect of processor speed, the time limit will be scaled for each individual processor. Details on the solution scoring follow.

8.1 Computational environment

Participants will perform IRP algorithm evaluation using their own computers. **Solvers must run on a single thread.** If a solver solves linear relaxations or mixed integer linear programming problems, any commercial solver may be used as long as the commercial solver is run on a single thread.

A time limit will be set for the solution of each instance. That limit will be standardized to take into account different machines and platforms according to the CPU marks provided by PassMark Single Thread Performance[1]. Runs must be performed on a processor listed in PassMark.

Currently, the top single thread CPU mark is 3,730, but the test results are updated daily and change frequently. Mid-range desktop processors have marks around 2,000. For this reason, we will use the mark 2,000 to define our standardized times. This means that if a run is performed on a processor (say AMD Ryzen Threadripper 2920X) that has mark 2,500, the time allowed will be scaled by 0.8, allowing less time than on a "Standard Processor".

8.2 Time limit

The time limit for each instance is to be 30 minutes on a processor with a CPU mark of 2,000.

If we let C denote the CPU mark for a given computer, the scaling factor associated with that computer is S, where $S = 2,000/C$. The time limit allowed for each instance is $S \times 30$ minutes. Thus, faster CPUs are afforded less time and slower ones more time.

This time limit ($S \times 30$ minutes) must be set for each instance. **All times are wall clock time**. It is up to the competitor to perform runs on a machine that is not heavily loaded.

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

9 Scoring solver performance

Let BKS be the value of the best known solution for a given instance and v be the solution found by the solver when the time limit was reached. The solver's score on that instance is calculated as:

$$SCORE = 100 \times \left(\frac{v}{BKS} - 1 \right). \tag{1}$$

A threshold of $1.1 \times BKS$ will be set on solution values v. Solution values exceeding this threshold will be set to $1.1 \times BKS$. The same value will be attributed to infeasible solutions or failure to provide a solution. In all of these cases, the corresponding score will be 10, which is the worst possible score.

If the solver finds a solution that is better than the best known solution, it is possible achieve a negative score.

The scores of individual instances are then converted into points: for each instance tested, all competing solvers are ranked according to their individual score. The best-scoring solver gets 10 points, the second 8, then 6, 5, 4, 3, 2, 1. Solvers that are not among the top 6 do not receive any points for that instance. In case of ties, the points at play are evenly split among the solvers involved. For example, 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 points for a solver is the sum of its points over all test instances. The competitor rankings will be based on total points, ties being broken by the average SCORE over all test instances.

The top THREE competing solvers will be tentatively designated as the IRP "finalists."

10 Article Submission and Invitations to Present at the Workshop

All competing teams should submit a short article describing their solver and results. These articles may be heavily based on work that is published elsewhere as long as this is clearly indicated.

Articles are limited to six pages, not counting possible appendices with detailed tables of results. An optional template will be available for download from the Implementation Challenge website.

The submitted articles will help the committee select which teams will be invited to present at the workshop (though we hope all competitors will be able to attend). Final versions of the articles will also be posted to the workshop website to share information among participants and with the community more broadly.

There are two pathways to being invited to present at the workshop:

- The tentative Finalists will be automatically invited to present, *provided that the article submitted adequately describes the methods used by their* *solver*. (Failure to submit an adequate description will forfeit the finalist designation and associated invitation to present at the workshop.)

- Additional competitors will also invited to present at the workshop. Selection will be based on algorithmic performance, as well as potentially broader criteria such as originality, strong performance on multiple variants, simplicity of approach, clarity of exposition, etc. (In addition to the finalists for each variant, we expect to select 10-15 additional papers across all variants for presentation.)

Solvers competing in multiple variants may choose to submit only one article. With permission from the Challenge committee, such articles may be slightly longer.

11 Journal Submissions

Challenge competitors are encouraged to submit a full article to one the journal special issues associated with the Implementation Challenge. Such articles should be of substantially new content and not published elsewhere. All submissions will be subject to the journal's usual review process and must meet its standards for publication through this process. There is no guarantee that submitted articles will be accepted for publication.

12 IRP Competition Schedule

The IRP competition is officially underway. The relevant dates for the IRP competition are:

October 5, 2020 – Release of the initial version of the IRP competition rules. This launched the IRP competition and allowed competitors to begin testing their solvers.

December 1, 2021 – Release of the final version of the IRP competition rules. (Only very minor changes are expected afterward).

December 8, 2021 – Deadline for competitors to register for the competition.

December 15, 2021 – Official list of competitors posted.

January 16, 2022 – Deadline for competitors to send all output files.

January 23, 2022 – Competition results are posted.

February 1, 2022 – Deadline for competitors to send documents in article format.

February 15, 2022 – Invitations to present at the workshop are sent.

March 15, 2022 – Deadline for competitors to submit the final versions of articles to post.

April 6-8, 2022 – 12th DIMACS Challenge Workshop.