AiSoft Challenge 2024:
Inventory Routing Problem (IRP)
Problem Description and Rules

Shiraz University

September 28, 2024

1 Problem Description

Inspiring from the implementation challenge held by DIMACS in 2020-2022, the first AiSoft challenge addresses the Inventory Routing Problem (IRP) which combines the process of inventory management with routing decisions. Due to its applications in supply chain management and integrated logistics, there has been a surge of interest in the IRP especially over the past decade (Archetti et al., 2007, 2017; Vadseth et al., 2021; Archetti et al., 2021; Skålnes et al., 2023; Laganà et al., 2024).

This problem is defined on a complete graph $G = (V, E)$ where $V = C \cup \{0\}$ is the set of vertices and $E = \{(i, j)|i, j \in V\}$ is the set of edges. Node 0 represents a central supplier, also referred to as the depot, and has a homogeneous fleet of m vehicles. The capacity of each vehicle is denoted by Q, and the vehicles are responsible for serving all customers in set $C = \{1, 2, ..., n\}$ within a given time horizon consisting of H periods (i.e., $T = \{1, 2, ..., H\}$). Each node $i \in V$ (i.e., the depot and each customer) has an inventory with minimum and maximum capacity of L_i and U_i, respectively. I_{i0} represents the starting inventory level of each node $i \in V$. In each period $t \in T$, the supplier produces r_{0t} products, known as the production rate. Moreover, r_{it} indicates the demand (i.e., consumption rate) of customer $i \in C$ in period $t \in T$. In each period, customer $i \in C$ must satisfy its demand without stuck-outing. To do so, using the available vehicles, the supplier must deliver its products to the customers in advance concerning the inventory levels. All vehicles must begin and end their paths at the depot. Each customer can be visited at most once in each period. According to the maximum-level policy, customers can receive any amount of goods during each visit, as long as the maximum inventory level is not exceeded.

Given c_{ij} as the traveling cost from node $i \in V$ to node $j \in V$, the goal is to build a distribution plan for the considered time horizon with minimum cost. The cost includes traveling costs along with inventory costs at the customers' locations and the supplier. To sum up, the IRP involves answering three major questions as follows:

- When to visit each customer
- How much to deliver at each visit

- Which routes the vehicles must perform

1.1 Basic Concepts

- **Depot:** a central supplier responsible for serving the customers using a limited number of homogeneous vehicles
- **Customer:** a retailer with a specific consumption rate in each period and an inventory with a pre-defined maximum and minimum capacity
- **Node:** both the central supplier (i.e., depot) and customers
- **Time horizon:** a pre-specified duration of time divided into H discrete periods
- **Inventory level:** the amount of products available in the inventory
- **Consumption rate:** the demand of each customer in each period
- **Production rate:** the amount of products that the supplier produces throughout each period

1.2 Constraints

The distribution plan is subject to the following constraints:

- The minimum and maximum inventory capacity of each node cannot be violated
- No stock-outs are allowed
- Each vehicle can perform at most one route per period
- Each customer can be visited at most once per period
- Each route starts and ends at the depot
- Vehicle capacity cannot be violated

1.3 Objective Function

In the IRP, the goal is to minimize the distribution cost which is the sum of the transportation cost and the expense of keeping products in the inventories belonging to both the supplier and the customers. The transportation cost is determined by the cost of traveling between each pair of vertices. The traveling cost between each pair of vertices is specified in advance considering traffic information, fuel cost, and other relevant information. Moreover, the inventory cost per unit of product for the supplier and each customer is also determined. For each customer $i \in C$, the inventory level in period $t \in T$ is the inventory level from period $t-1$ plus the amount delivered in period t minus the amount consumed in period t. Thus, we have $I_{it} = I_{i(t-1)} + q_{it} - r_{it}$. Note that inventory costs are also considered for initial inventory levels.

2 Instance Files

Each problem instance is provided in a single file with a fixed structure that can be deduced from the following example.

```
nVehicles:  10
nDepots:     1
nCustomers: 10
nPeriods:    2
VEHICLE_TYPE    FLEET_SIZE     CAPACITY
           1            10            8
DEPOT          X       Y       INV_COST       INITIAL_INV    MIN_LEVEL_INV   MAX_LEVEL_INV   PRODUCTION_0    PRODUCTION_1
       0       428     298            0                99                0             622             70              70
CUSTOMER       X       Y       INV_COST       INITIAL_INV    MIN_LEVEL_INV   MAX_LEVEL_INV   DEMAND_0        DEMAND_1
       1       211     283         0.22                 7                0              31        5               8
       2       232     365         0.22                 1                0              54        3               6
       3       223     318         0.05                 8                0              14        3               5
       4       234     325         0.16                 6                0              69        6               9
       5       254     325         0.06                 3                0              23        4               4
       6       295     428         0.02                 3                0              47        0               0
       7       181     490         0.19                 9                0              32        6               8
       8       110     177         0.08                11                0              56        6              10
       9        71     236         0.21                 5                0              21        4               5
      10       459     143         0.21                10                0              27        4               6
COST_MATRIX
          0       1       2       3       4       5       6       7       8       9      10
  0
  1     109
  2     104      43
  3     103      19      24
  4      98      24      21       7
  5      89      31      23      16      11
  6      93      84      45      66      60      56
  7     157     105      68      89      87      91      65
  8     171      74     113      91      97     104     156     161
  9     182      74     104      87      93     102     148     139      36
 10      80     143     159     147     145     138     165     223     176     200
```

Figure 1: An example of a data file

The first four lines determine the number of vehicles, depots, customers, and periods. In the next two lines, the capacity of each vehicle is shown along with the type and number of vehicles. In each instance, vehicles are assumed to be homogeneous in terms of their types. Thus, they also have similar maximum capacity. As explained before, the IRP with only a single depot is assumed in this challenge. Thus, in all instances, the value of 'nDepots' is 1. Line 8 presents the characteristics of the depot node where

- $DEPOT$ = node number corresponding to the supplier (depot)
- X = x coordinate of the supplier
- Y = y coordinate of the supplier
- INV_COST = unit inventory cost at the supplier
- $INITIAL\_INV$ = starting level of the inventory at the supplier
- MIN_LEVEL_INV = minimum level of the inventory at the depot
- MAX_LEVEL_INV = maximum level of the inventory at the depot

- $PRODUCTION\_i$ = quantity of product made available at the supplier at each discrete time instant of the planning time horizon

After identifying the characteristics of the common supplier, each of the n customers is described in the subsequent n lines. For each customer $i \in C$, the following information is provided:

- $CUSTOMER$ = customer number
- X = x coordinate of customer i
- Y = y coordinate of customer i
- INV_COST = unit inventory cost at customer i
- $INITIAL\_INV$ = starting level of the inventory at customer i
- MIN_LEVEL_INV = minimum level of the inventory at customer i
- MAX_LEVEL_INV = maximum level of the inventory at customer i
- $DEMAND\_i$ = the quantity used by customer i at each discrete time instant of the planning time horizon

At the end of the data file, the transportation costs (i.e., c_{ij} for each two nodes i and j) are represented as a diagonal matrix.

3 Solution Files

For each instance, a solution file must be provided which includes the best feasible solution found for that instance. Each solution file must be named 'Solu-InsName.txt' where 'InsName' is the name of the associated instance file without extension. For example, the output file associated with the instance 'Sabs1n20H3.dat' would be 'Solu-Sabs1n20H3.txt'. Besides, the content of this file has to have a specific format, otherwise, your algorithm will take the last rank of the sample.

In the following, considering 10 customers, three vehicles, and two periods, an IRP solution is presented as an example. For each period, the paths of the vehicles and the total distribution cost of that period are shown in a single line. These paths are separated from each other using brackets. Each bracket consists of a list of paired values $(CNum, Del)$, where $CNum$ represents the customer number, and Del represents the quantities delivered to that customer. The next three lines demonstrate the total transportation cost and inventory cost separately. The total cost of the presented solution is reported in the last line.

Period 1: [(9,2)(8,3)(7,3)] [(6,1)(4,3)(5,1)(2,3)] cost: 406
Period 2: [(5,7)(9,1)] [(10,2)(7,3)(1,1)(3,1)] [(2,1)] cost: 1041
Total transportation cost: 1000
Total inventory cost at customer locations: 21

Total inventory cost at the depot: 20
Total cost: 1447

Please note that each route must begin and end at the depot. But, for simplicity, node 0 is not explicitly shown in the solution file. There is a space between each pair of brackets and between the last bracket in each line and the term 'cost'. Additionally, in the mentioned format, there is always a space between ':' and its subsequent value. For instance, we have: 'Period space 1: space [(9,2)(8,3)(7,3)] space [(6,1)(4,3)(5,1)(2,3)] space cost: space 406' or we have 'Total cost: space 1447'. Output files will be automatically read and processed, and any files that do not comply with the specified format will be discarded.

4 Primary Competition Rules

Rule 1: Participants must have academic affiliation and are allowed to work in groups of a maximum of three members.
Rule 2: Participants must implement a solving technique to tackle the problem on a single thread. They can use any programming language that runs under Windows or Linux. Any open-source libraries may be used as long as running on a single thread.
Rule 3: A dataset of 10 instances with different sizes is available from the opening day for the contestants' experiments. Another 20 instances are going to become available by the end of this month which are used to determine the finalists. A third dataset will be used to rank the finalists. The instances belonging to the three datasets are called *Public1*, *Public2*, and *Hidden* instances, respectively.
Rule 4: The algorithms must take an input instance file in the given format, and produce an output solution file in the described format. The same version of the algorithm must be used for all instances.
Rule 5: The algorithm can be either deterministic or stochastic, but the results must be reproducible. In particular, the participants that use a stochastic algorithm should code their program so that each submitted solution can be reproducible (for example, by setting a fixed random seed).
Rule 6: It is recommended to propose an anytime algorithm in which once a better solution is found, the old solution file is replaced by the new one. In this way, you can be more sure to have at least an output file before the time ends.
Rule 7: The set of at most 10 finalists will be selected after the competition deadline (Ranked according to their results on the Public2 instances). Ranking of the finalists will be based on the scores provided on the Hidden instances. The ranking will be based on the average ranking of each contestant in each instance.

5 Competition Schedule

The relevant dates for this competition are:

- **September 28, 2024 - Release of the description of the problem and the initial competition rules:** This launched the IRP competition and allowed competitors to begin working on their solvers using the first part of the public instances.

- **October 19, 2024 - Release of the final version of the competition rules and the rewards:** Only very minor changes in the rules are expected afterward. The second part of the public instances are also provided which are used to determine the finalists.
- **December 7, 2024 - Deadline for competitors to send all output files**
- **December 10, 2024 - Competition results are posted and finalists are announced**
- **December 14, 2024 - Final ranking announcement:** Invitations to present at the workshop are sent.
- **December 24-26, 2024 - Holding the 1^{st} AiSoft Challenge Workshop**

References

Archetti C, Bertazzi L, Laporte G, Speranza MG. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation science 2007;41(3):382–391.

Archetti C, Boland N, Grazia Speranza M. A matheuristic for the multivehicle inventory routing problem. INFORMS Journal on Computing 2017;29(3):377–387.

Archetti C, Guastaroba G, Huerta-Muñoz DL, Speranza MG. A kernel search heuristic for the multivehicle inventory routing problem. International Transactions in Operational Research 2021;28(6):2984–3013.

Laganà D, Malaguti E, Monaci M, Musmanno R, Paronuzzi P. An iterated local search matheuristic approach for the multi-vehicle inventory routing problem. Computers & Operations Research 2024;p. 106717.

Skålnes J, Vadseth ST, Andersson H, Stålhane M. A branch-and-cut embedded matheuristic for the inventory routing problem. Computers & Operations Research 2023;159:106353.

Vadseth ST, Andersson H, Stålhane M. An iterative matheuristic for the inventory routing problem. Computers & Operations Research 2021;131:105262.