Benchmark Set for the IEEE WCCI-2020 Competition on Evolutionary Computation for the Electric Vehicle Routing Problem

Michalis Mavrovouniotis, Charalambos Menelaou,
Stelios Timotheou, Christos Panayiotou, Georgios Ellinas,
Marios Polycarpou

December 13, 2019

KIOS Research and Innovation Center of Excellence,
Department of Electrical and Computer Engineering,
University of Cyprus,
Nicosia, Cyprus

{mavrovouniotis.michalis,menelaou.charalambos,timotheou.stelios, christosp,gellinas,mpolycar}@ucy.ac.cy

1 Introduction

Transportation has been the main contributor to CO_2 emissions. Due to global warming, pollution and climate changes, logistic companies such as FedEx, UPS, DHL and TNT have became more sensitive to the environment and they are investing in ways to reduce the CO_2 emissions that result as part of their daily operations. There is no doubt that using electric vehicles (EVs) instead of conventional vehicles will significantly contribute to the reduction of CO_2 emissions [1].

With the growing interest of logistic companies in utilizing EVs for their daily operations, a problem of routing a fleet of EVs has emerged, namely the electric vehicle routing (EVRP) [2]. In this report, a benchmark set of the EVRP is provided with known and unknown optimum values. The rest of the paper is organized as follows. Section 2 gives a description of the EVRP. Section 3 gives details of the benchmark set. Section 4 gives the criteria of evaluating an algorithm on the benchmark set.

2 The Electric Vehicle Routing Problem

The EVRP is a challenging $\mathcal{NP}$-hard combinatorial optimization problem. It can be described as follows: given a fleet of EVs, we need to find the best possible routes within the battery charge level limits, starting and ending to the central depot, to serve a set of customers.

Usually, the problem is represented by a fully connected weighted graph $G = (V, A)$, where $V = \{I \cup 0 \cup F'\}$ is a set of nodes and $A = \{(i, j) \mid i, j \in V, i \neq j\}$ is a set of arcs connecting these vertices. I denotes the set of customers, $\{0\}$ denotes the central depot, and F' denotes the set of β_i vertex copies of each charging station $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$), to permit multiple visits to each vertex in F [3]. With each arc, a non-negative value distance d_{ij} is associated which represents the distance between vertices i and j. Each traveled arc (i, j) consumes the amount $h \cdot d_{ij}$ of the remaining battery charge of the EV traveling the arc, where h denotes the constant charge consumption rate of the EV. A fleet of fully charged homogeneous EVs with a maximal capacity of Q is positioned at the depot. Each vertex $i \in V$ is assigned a positive demand q_i which is 0 if $i \notin I$. Variable u_i specifies the remaining cargo and y_i the remaining battery charge on arrival at vertex $i \in V$ of the EV, which are initially set to $u_0 = Q$ and $y_0 = C$, respectively, where C is the maximal battery level.

Formally, the EVRP can be described as follows:

$$\min \sum_{i \in V, j \in V, i \neq j} d_{ij} x_{ij}, \tag{1}$$

s.t

$$\sum_{j \in V, i \neq j} x_{ij} = 1, \forall i \in I, \tag{2}$$

$$\sum_{j \in V, i \neq j} x_{ij} \leq 1, \forall i \in F', \tag{3}$$

$$\sum_{i \in V, i \neq j} x_{ji} - \sum_{i \in V, i \neq j} x_{ij} = 0, \forall j \in V \backslash \{0\}, \tag{4}$$

$$0 \leq u_j \leq u_i - q_i x_{ij} + C(1 - x_{ij}), \forall i \in V, \forall j \in V, i \neq j, \tag{5}$$

$$0 \leq u_0 \leq C, \tag{6}$$

$$0 \leq y_j \leq y_i - (h \cdot d_{ij}) x_{ij} + Q(1 - x_{ij}), \forall i \in I, \forall j \in V, i \neq j, \tag{7}$$

$$0 \leq y_j \leq Q - (h \cdot d_{ij}) x_{ij}, \forall i \in F' \cup \{0\}, \forall j \in V, i \neq j, \tag{8}$$

$$x_{ij} \in \{0, 1\}, \ \forall i \in V, \ \forall j \in V, i \neq j, \tag{9}$$

where Eq. (1) defines the EVRP objective function, Eq. (2) enforce the connectivity of customer visits, Eq. (3) handles the connectivity of visits to recharging stations, Eq. (4) establish flow conservation by guaranteeing that at each vertex, the number of incoming arcs is equal to the number of outgoing arcs, Eq. (5) and Eq. (6) guarantee demand fulfillment at all customers by assuring a non-negative cargo load upon arrival at any vertex including the depot, Eq. (7) and Eq (8) ensure that the battery charge never falls below 0, and Eq. (9) is a binary decision variable which is equal to 1 if an arc is traveled and 0 otherwise.

3 Description of EVRP Benchmark Set

The EVRP benchmark set consists of 7 problem instances (up to 100 customers) with calculated optimums (or upper bounds) and 9 larger problem instances (up to 1000 customers) with unknown optimums. The first set of EVRP instances was generated using the well-known instances of the conventional vehicle routing problem from Christofides and Eilon [4] (see Fig. 1) and the second set from the recent instances of the conventional vehicle routing problem from Uchoa *et al.* [5] (see Fig. 2). The small problem instances of the first set are useful for testing (e.g., validation of the solver, parameter tuning, etc.), since the large problem instances are more challenging and time-consuming to solve. The details of all the generated EVRP instances are summarized in Table 1. The columns in Table 1 present the number of customers, the number of depots, the number of charging stations, the minimum number of routes, the maximum load of an EV, the maximum battery charge level of an EV, the energy consumption constant, and an upper bound value (it could be optimal in some cases but it is not verified yet).

The file of each EVRP instance of the benchmark set contains the following keywords:

- `COMMENT`: information about the problem instance
- `OPTIMAL_VALUE`: the optimal value (or upper bound) of the problem instance (if known; otherwise is set to 0)
- `VEHICLES`: minimum number of EVs (or routes)
- `DIMENSION`: the number of nodes including the central depot
- `STATIONS`: the number of charging stations
- `CAPACITY`: the maximum cargo capacity of the EV (i.e., C)
- `ENERGY_CAPACITY`: the maximum battery charge of the EV (i.e., Q)
- `ENERGY_CONSUMPTION`: the constant charge consumption rate (i.e., h)
- `EDGE_WEIGHT_FORMAT`: euclidean distance
- `NODE_COORD_SECTION`: this section contains the information of the nodes, in the format of node id, x and y coordinates

Table 1: Details of the EVRP benchmark set

name	#customers	#depots	#stations	#routes	C	Q	h	UB
E-n22-k4.evrp	21	1	8	4	6000	94	1.2	384.67
E-n23-k3.evrp	22	1	9	3	4500	190	1.2	573.13
E-n30-k3.evrp	29	1	6	4	4500	178	1.2	511.25
E-n33-k4.evrp	32	1	6	4	8000	209	1.2	869.89
E-n51-k5.evrp	50	1	5	5	160	105	1.2	570.17
E-n76-k7.evrp	75	1	7	7	220	98	1.2	723.36
E-n101-k8.evrp	100	1	9	8	200	103	1.2	899.88
X-n214-k11.evrp	213	1	9	11	944	987	1.0	–
X-n352-k40.evrp	351	1	35	40	436	649	1.0	–
X-n459-k26.evrp	458	1	20	26	1106	929	1.0	–
X-n573-k30.evrp	572	1	6	30	210	1691	1.0	–
X-n685-k75.evrp	684	1	25	75	408	911	1.0	–
X-n749-k98.evrp	748	1	30	98	396	790	1.0	–
X-n819-k171.evrp	818	1	25	171	358	926	1.0	–
X-n916-k207.evrp	915	1	9	207	33	1591	1.0	–
X-n1001-k43.evrp	1000	1	9	43	131	1684	1.0	–

- `DEMAND_SECTION`: this section contains the demands of each customer, in the format of node id and demand (i.e., q_i)
- `STATIONS_COORD_SECTION`: this section contains the node id of the charging stations
- `DEPOT_SECTION`: this section contains the node id of the central depot

The sample code available in the competition website[1] is able to read all the aforementioned information and they can be utilized using the corresponding functions provided. Example of utilizing the `EVRP.hpp` implementation to read the problem the benchmark problem and evaluate the solver, as well as an example of utilizing the `stats.hpp` implementation to store and output the results of the solver are provided. The `main.cpp` provided can be used by replacing the `initialize_random_heuristic()` and `generate_random_solution()` with the corresponding implementation of your solver. The benchmark set is also available from the same website.

4 Evaluation Criteria

- **Problem Instances**: The 16 EVRP problem instances are summarized in Table

[1] `https://mavrovouniotis.github.io/EVRPcompetition2020/`

1

- **Independent Runs**: 20 (with seeds from $1 - 20$)
- **Evaluations**: The maximum number of evaluations is $25000 \cdot n$ steps of $\mathcal{O}(n^2)$, where $n = |I| + 1 + |F'|$
- **Termination Condition**: When the algorithm reaches the maximum number of evaluations (in other words calling the objective function).
- **Measurement**: The best solution found from all evaluations as follows:

$$\bar{P} = \frac{1}{R}\sum_{i=1}^{R} P_i^*, \tag{10}$$

where R is the number of independent runs (i.e., $R = 20$), and P_i^* is the best solution found from all evaluations in run i.

NOTE: The $\bar{P}$ measurement is already implemented in the sample code and stored in text files. You can simply submit the text files obtained for each instance together with the details and source code of your algorithm. Table 2 shows an example of the results obtained from the heuristic implemented in the sample code, in which the “mean” is the average of the 30 runs, “stdev” is the standard deviation, “min” is the best result of the 30 runs, and “max” is the worst result of the 30 runs. All these values are calculated in the output text files of the source code.

5 Conclusion

In this report we have proposed a set of 15 EVRP benchmark instances to evaluate algorithms. The EVRP benchmark instances impose new challenges to the ordinary VRP problem since algorithms have to consider the possibility of de-routing to visiting a charging station for recharging while serving all the demands of the customers. The primary goal in generating this set of benchmark instances is to boost the research on the applications of the EVRP.

References

[1] E. Commission. (2009) Effort sharing: Member states emission targets. [Online]. Available: https://ec.europa.eu/clima/policies/effort_en

[2] F. Gonçalves, S. Cardoso, and S. Relvas, “Optimization of distribution network using electric vehicles: A VRP problem,” University of Lisbon, Tech. Rep. CEG-IST, 2011.

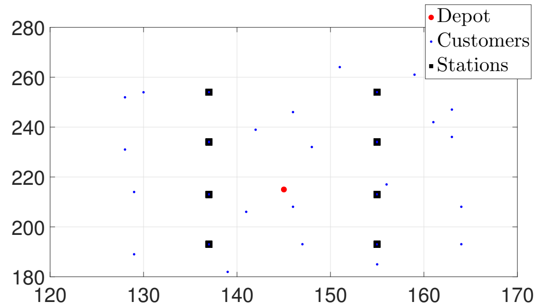
Table 2: Example – Random Heuristic results.

id	problem instance	$\bar{P}$			
		mean	stdev	min	max
1	`E-n22-k4.evrp`	621.78	17.9	583.2	660.5
2	`E-n23-k3.evrp`	1037.14	35.1	961.4	1091.8
3	`E-n30-k3.evrp`	1129.5	36.6	1039.0	1196.3
4	`E-n33-k4.evrp`	1377.3	30.5	1303.4	1428.5
5	`E-n51-k5.evrp`	1550.6	23.9	1498.0	1592.6
6	`E-n76-k7.evrp`	2647.8	43.8	2531.3	2711.5
7	`E-n101-k8.evrp`	3707.1	40.8	3613.3	3808.2
8	`X-n214-k11.evrp`	59075.0	601.1	57594.8	60019.4
9	`X-n352-k40.evrp`	164585.2	1117.9	161915.3	166531.5
10	`X-n459-k26.evrp`				
11	`X-n573-k30.evrp`				
12	`X-n685-k75.evrp`				
13	`X-n749-k98.evrp`				
14	`X-n819-k171.evrp`				
15	`X-n916-k207.evrp`				
16	`X-n1001-k43.evrp`				

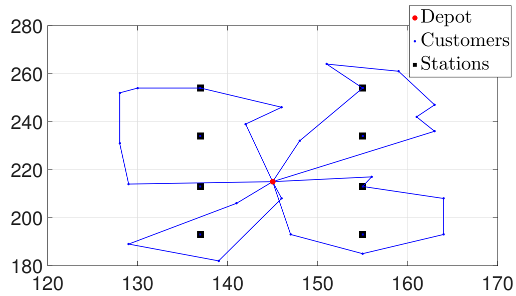
[3] S. Erdoğan and E. Miller-Hooks, "A green vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 100 – 114, 2012.

[4] N. Christofides and S. Eilon, "An algorithm for the vehicle-dispatching problem," *OR*, vol. 20, no. 3, pp. 309–318, 1969.

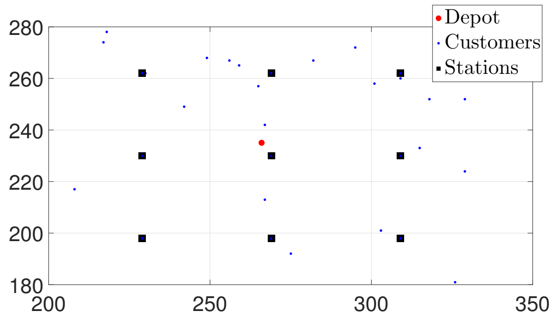
[5] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 257, no. 3, pp. 845 – 858, 2017.


Depot
Customers
Stations
280
260
240
220
200
180
120
130
140
150
160
170

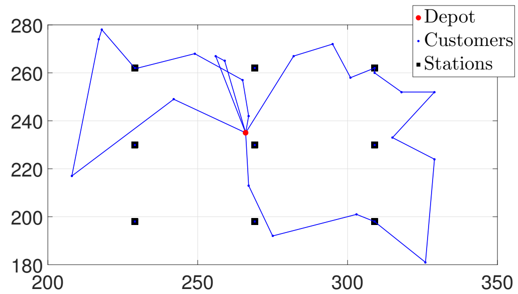
(a)


Depot
Customers
Stations
280
260
240
220
200
180
120
130
140
150
160
170

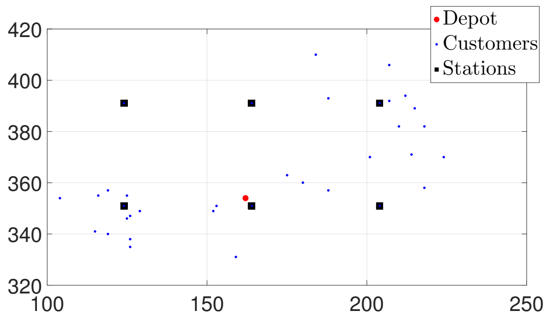
(b)


Depot
Customers
Stations
280
260
240
220
200
180
200
250
300
350

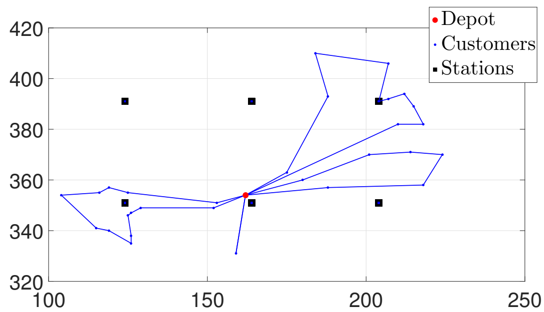
(c)


Depot
Customers
Stations
280
260
240
220
200
180
200
250
300
350

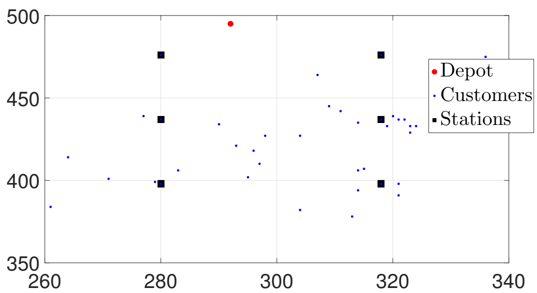
(d)


Depot
Customers
Stations
420
400
380
360
340
320
100
150
200
250

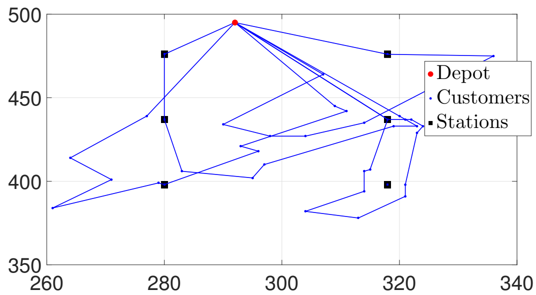
(e)


Depot
Customers
Stations
420
400
380
360
340
320
100
150
200
250

(f)


Depot
Customers
Stations
500
450
400
350
260
280
300
320
340

(g)


Depot
Customers
Stations
500
450
400
350
260
280
300
320
340

(h)

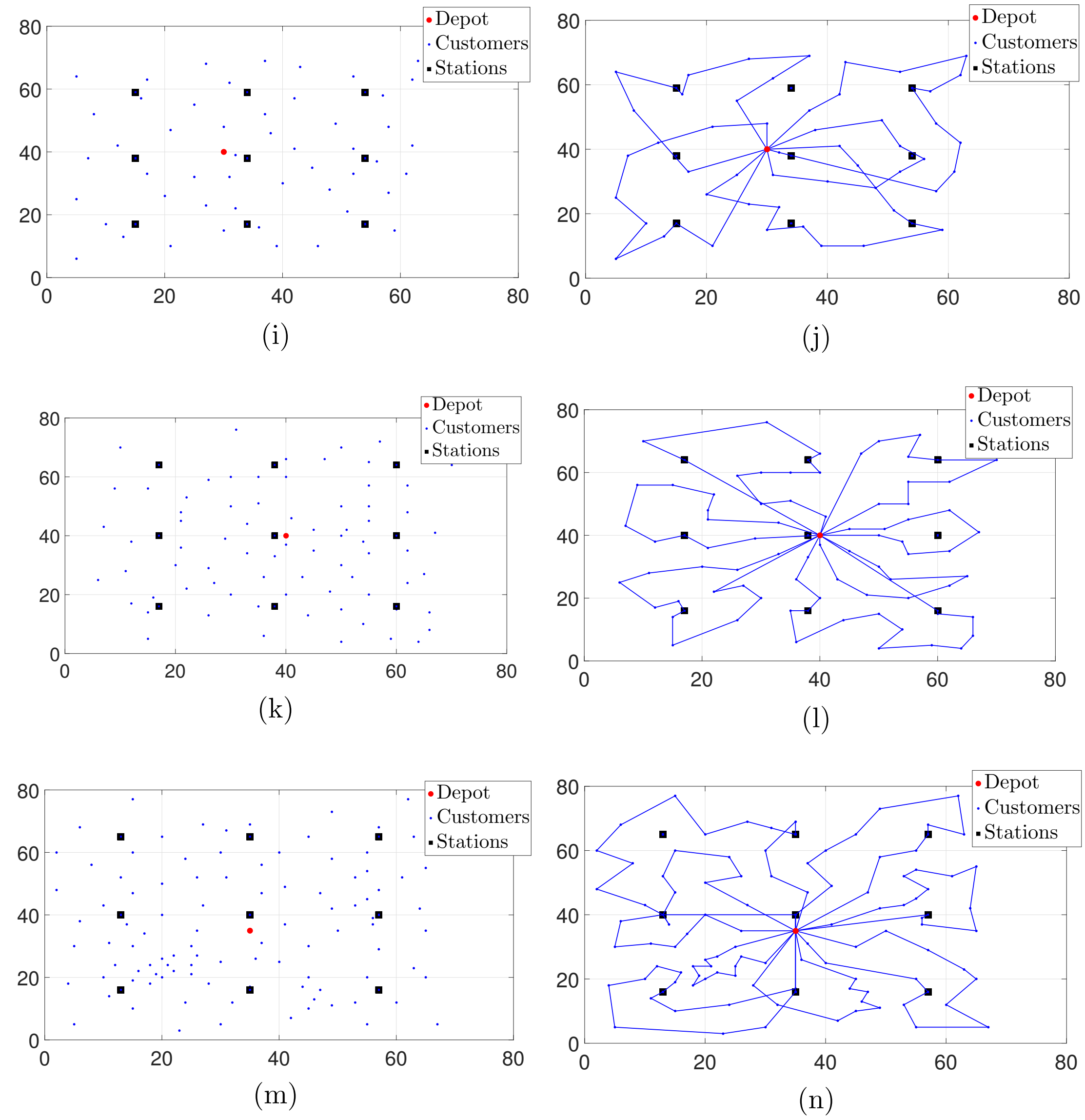


Figure 1: Illustration of problem instances (left) with the known upper bound solution (right). These problem instances are useful for testing purposes and parameter tuning.

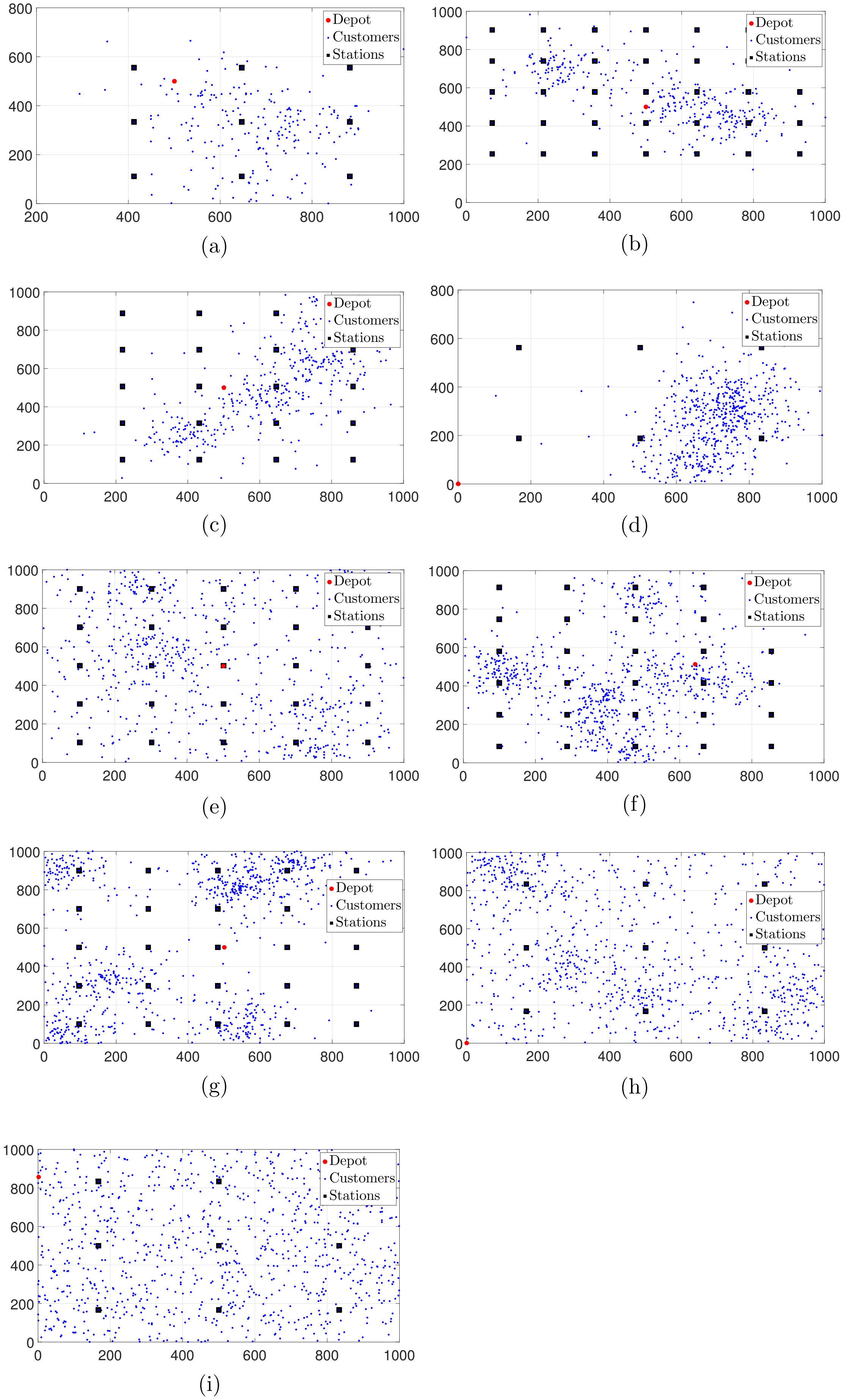


Figure 2: Illustration of problem instances with unknown upper bound.