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

March 18, 2020

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

**{mavrovouniotis.michalis,menelaou.charalampos,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], a fact that increases the interest of logistic companies in utilizing EVs for their daily operation.

In these circumstances, the problem of routing a fleet of EVs has emerged, namely the Electric Vehicle Routing Problem (EVRP) [2]. In this report, a benchmark set of EVRP instances is provided with known and unknown optimum values. The rest of this report is organized as follows. Section 2 presents the EVRP problem in which a detailed mathematical formulation of the problem is presented. Section 3 gives details of the EVRP benchmark set. Section 4 demonstrates the criteria of evaluating an algorithm on the benchmark set. Finally Section 5 concludes this report.

2 The Electric Vehicle Routing Problem

The EVRP is a challenging $\mathcal{NP}$-hard combinatorial optimization problem as it is an extension of the ordinary shorted path problem incorporating additional constraints [3]. The EVRP can be described as follows: given a fleet of EVs, we need to find the best possible route for each EV within their battery charge level limits, starting and ending to the central depot, to serve a set of customers.

Usually, the problem is expressed with the use of a fully connected weighted graph $G = (V, A)$, where $V = \{0 \cup I \cup F'\}$ is a set of nodes and $A = \{(i, j) \mid i, j \in V, i \neq j\}$ is a set of arcs connecting these nodes. Set I denotes the set of customers, set F' denotes the set of β_i node copies[1] of each charging station $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$), and 0 denotes the central depot. With each arc a non-negative value distance d_{ij} is associated which represents the euclidean distance between nodes i and j. Each traveled arc (i, j) consumes the amount hd_{ij} of the remaining battery charge level of the EV traversing that arc, where parameter h denotes the consumption rate of the EVs. Furthermore, each customer $i \in I$ is assigned a positive demand b_i[2].

The objective function of the EVRP, assuming a homogeneous fleet of EVs, is to find a set of routes that minimize the total distance traveled where:

- every customer is visited exactly once by exactly one EV
- all EVs begin (fully loaded and charged) and end at the depot
- for every EV route the total demand of customers does not exceed the EV's maximal carrying capacity C
- for every EV route the total energy consumption does not exceed the EV's maximal battery charge level Q
- EVs always leave the charging station fully charged (note that the depot is also considered as a charging station)
- the charging stations (including the depot) can be visited multiple times by any EV

Consequently, the EVRP can be mathematically formulated as follows:

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

s.t

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

[1] The node copies of stations are used to permit multiple visits to each charging station in a similar manner as proposed in [4]. An upper bound on the number of node copies for each charging station to consider is $\beta_i = 2|I|$, because at worst an EV for each customer is needed and a visit to a charging station before and after serving it [5].

[2] For the central depot and charging stations the demand is set as follows: $b_0 = 0 \wedge b_i = 0, \forall i \notin I$.

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

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

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

$$0 \leq u_i \leq C, \forall i \in V, \tag{6}$$

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

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

$$0 \leq y_i \leq Q, \forall i \in V, \tag{9}$$

$$x_{ij} \in \{0, 1\}, \ \forall i \in V, \ \forall j \in V, i \neq j, \tag{10}$$

where Eq. (1) defines the EVRP objective function, Eq. (2) enforce the connectivity of customer visits, Eq. (3) handles the connectivity of recharging stations, Eq. (4) establish flow conservation by guaranteeing that at each node, i.e., 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 carrying load upon arrival at any node including the depot, Eq. (7), Eq (8) and Eq. (9) ensure that the battery charge never falls below 0, and Eq. (11) define a set of binary decision variables which each one equal to 1 if an arc is traveled and 0 otherwise. Variables u_i and y_i denote, respectively[3], the remaining carrying capacity and remaining battery charge level of an EV on its arrival at node $i \in V$.

3 Description of EVRP Benchmark Set

The EVRP benchmark set consists of two groups of problems:

1. consists of 7 small problem instances (up to 100 customers) in which their optimal upper bound values are provided.
2. consists of 10 larger problem instances (up to 1000 customers) in which their upper bound values are not provided.

[3]These variables are initialized to $u_0 = C$ and $y_0 = Q$.

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-n143-k7.evrp`	142	1	4	7	1190	2243	1.0	–
`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	–

The first group of EVRP instances was generated by extending the well-known instances of the conventional vehicle routing problem from Christofides and Eilon [6] (see Figure 1) while the second group is an extension of the recent instances of the conventional vehicle routing problem from Uchoa *et al.* [7] (see Figure 2). The instances of the first group 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 (C), the maximum battery charge level of an EV (Q), the energy consumption constant (h), and an optimal upper bound (UB) 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) a solution can have

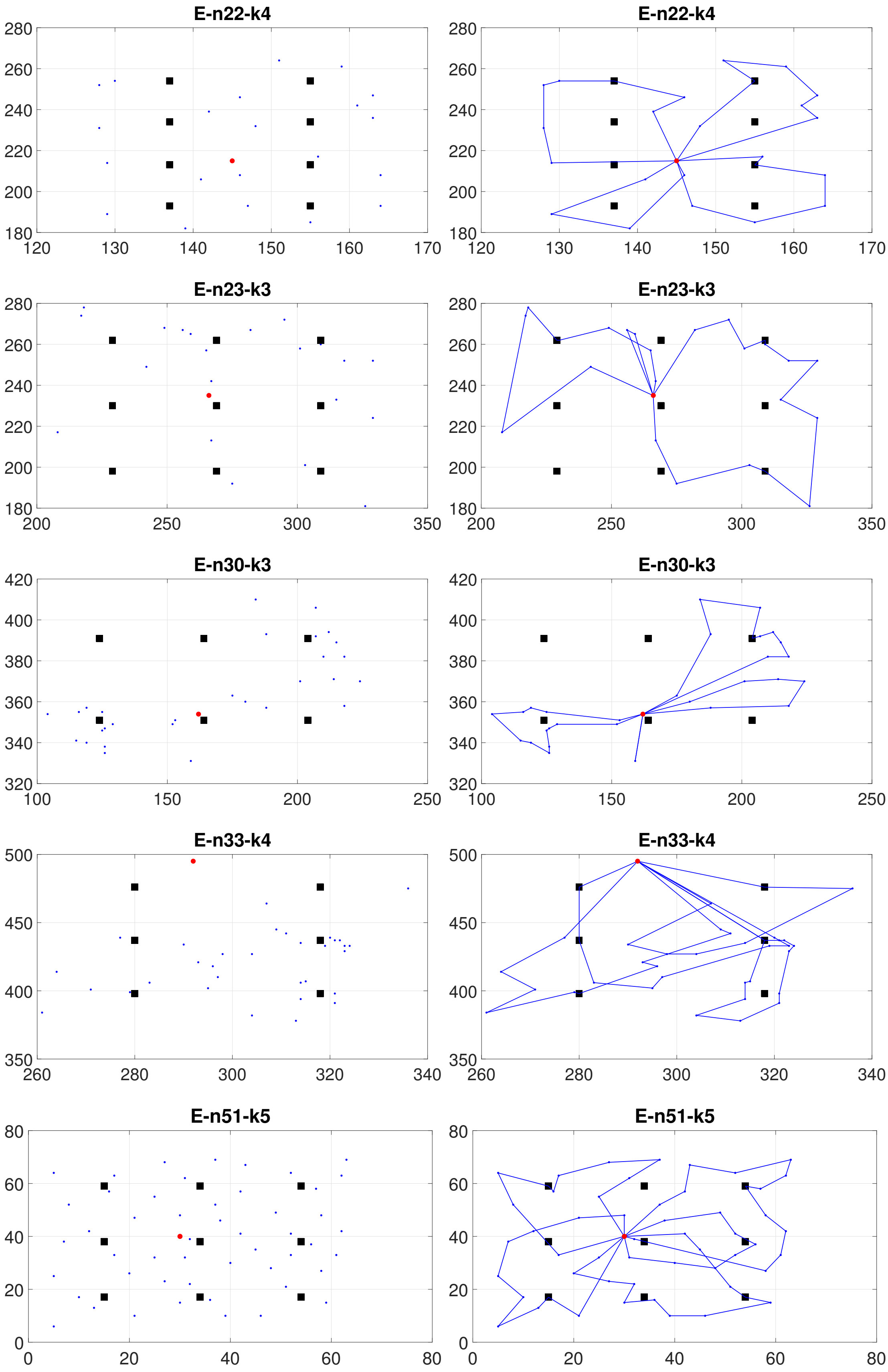

Figure 1: continued

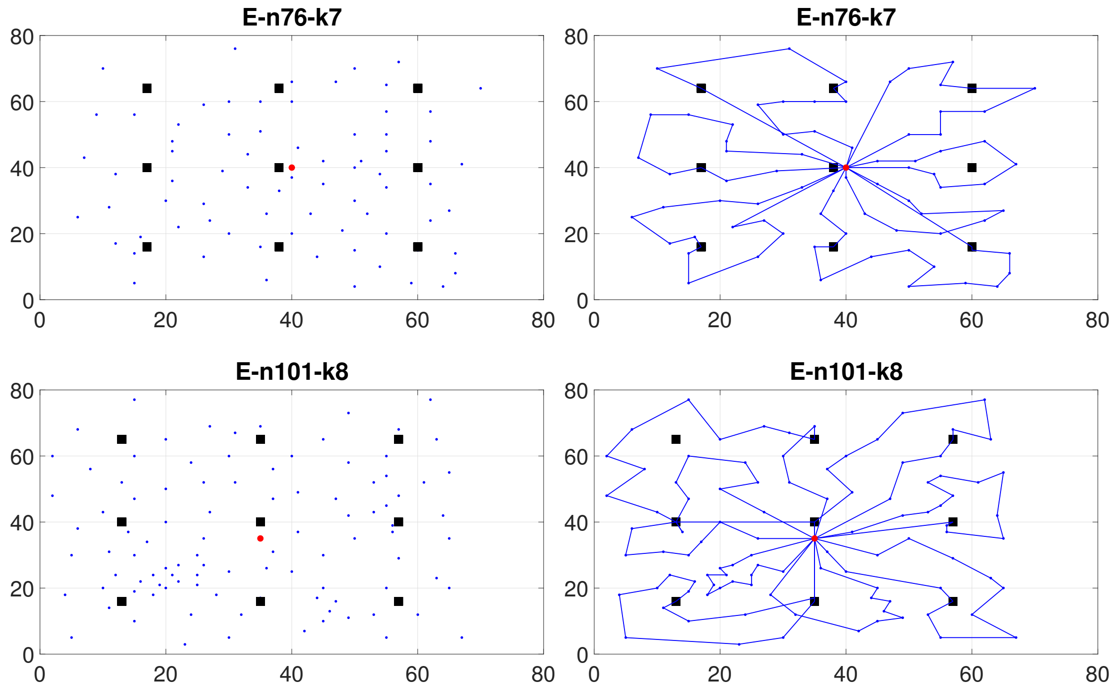

Figure 1: Illustration of problem instances (left) with the known upper bound solution (right). These problem instances are useful for testing purposes and parameter tuning. Note that the symbols: ●, ·, and ■ represent the depot, customers and charging stations respectively.

- `DIMENSION`: the number of customers including the central depot
- `STATIONS`: the number of charging stations
- `CAPACITY`: the maximum carrying capacity of the EV (i.e., C)
- `ENERGY_CAPACITY`: the maximum battery charge level 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
- `DEMAND_SECTION`: this section contains the demands of each customer, in the format of node id and demand (i.e., b_i)
- `STATIONS_COORD_SECTION`: this section contains the node ids of the charging stations
- `DEPOT_SECTION`: this section contains the node id of the central depot

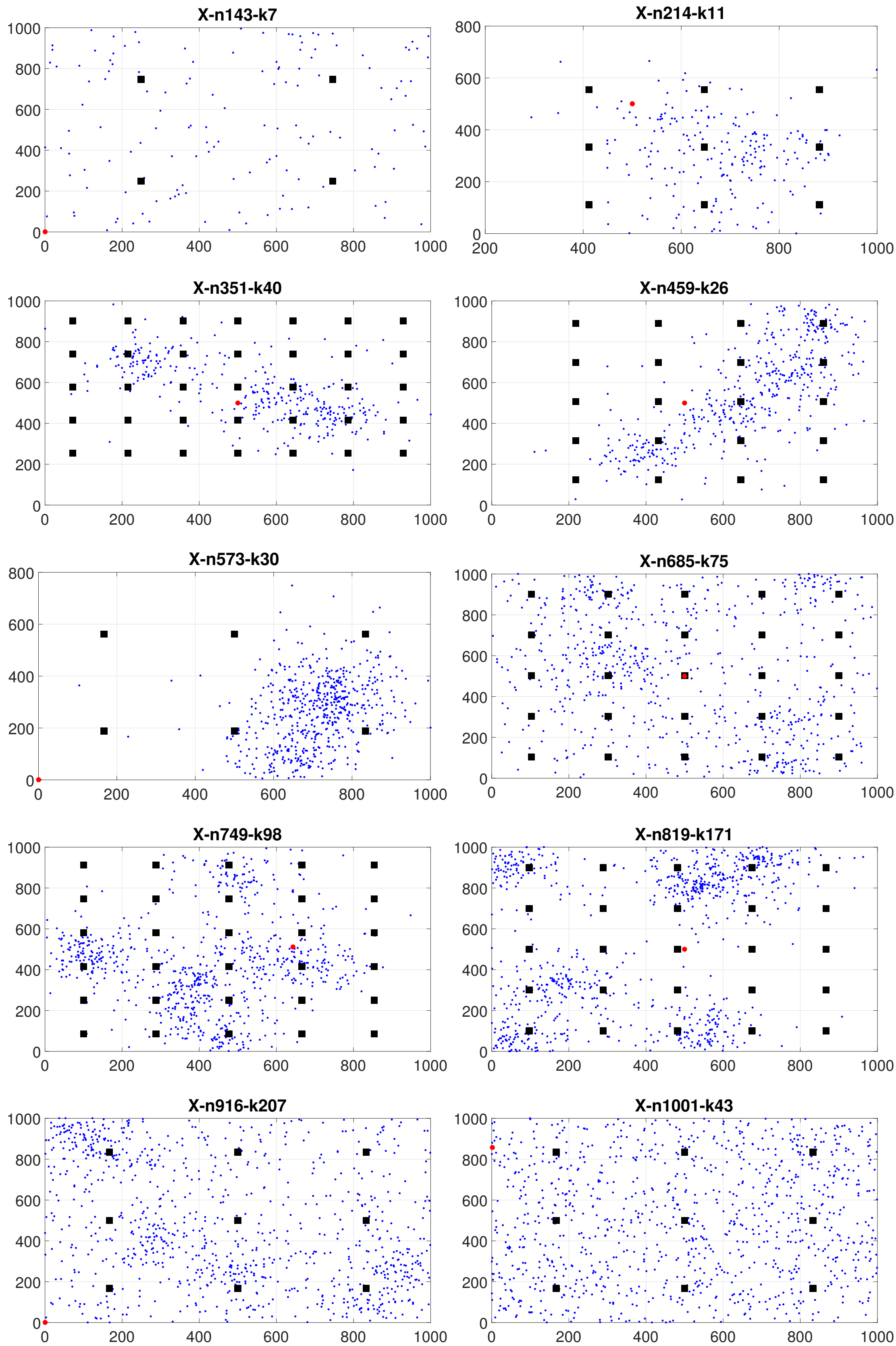

Figure 2: Illustration of problem instances with unknown upper bound. Note that the symbols: ●, ·, and ■ represent the depot, customers and charging stations respectively.

In the competition website[4] a sample source code that is able to read all the aforementioned information (i.e., read the files with extension `.evrp`) is provided in the file (`EVRP.hpp`) and can be utilized through the corresponding functions. More specific, the functions implemented in `EVRP.hpp` can be used to generate the distance matrix of the EVRP instance, access all the aforementioned information, and evaluate the solution generated by a solver (note that the solution must be in a specific format described in the source code). Additionally, the implementation of file `stats.hpp` provides functions that can be used to save the results of the solver that can be submitted to the competition. Participants are strongly encouraged to utilize the provided file `main.cpp` as it is and implement their approach in the `heuristic.cpp` file. Finally, the benchmark set described in Table 1 is also available at the completion website.

4 Evaluation Criteria

- **Problem Instances**: The 17 EVRP instances are summarized in Table 1
- **Independent Runs**: In case the implemented heuristic is of stochastic nature, then run 20 independent runs (with random seeds from $1 - 20$); otherwise run it once. Already defined in the sample code as `MAX_TRIALS`.
- **Evaluations**: The maximum number of evaluations is $25000n$, where $n = |I| + 1 + |F|$ is the size of the problem instance. One evaluation is of complexity $\mathcal{O}(n^2)$.
- **Termination Condition**: When the algorithm reaches the maximum number of evaluations defined above [in other words calling the objective function in Eq. (1)]. Already defined in the sample code as `TERMINATION`.
- **Measurement**: The best solution found from all evaluations averaged over multiple independent runs as follows:

$$\bar{P} = \frac{1}{R}\sum_{i=1}^{R} P_i^*, \tag{11}$$

  where R is the number of independent runs (i.e., $R = 20$ for stochastic approaches and $R = 1$ for deterministic approaches), 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 (in `stats.hpp`) and stored in output text files that are generated when the code is executed. Participants can simply submit these output text files obtained for each instance together with the details and source code of their algorithm. Table 2 shows an example of the results obtained from the sample heuristic implemented in the source code (file `heuristic.hpp`), in which the "mean" is the average solution quality of the 20 runs, "stdev" is the standard deviation, "min" is the best result of the 20 runs, and "max" is the worst result of the 20

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

Table 2: Example – Random Heuristic results.

id	name	$\bar{P}$			
		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-n143-k7.evrp`	78295.0	472.2	77983.2	79114.0
9	`X-n214-k11.evrp`	59075.0	601.1	57594.8	60019.4
10	`X-n352-k40.evrp`	164585.2	1117.9	161915.3	166531.5
11	`X-n459-k26.evrp`	215086.5	1522.0	212535.4	217665.8
12	`X-n573-k30.evrp`	180595.6	831.4	178858.2	182208.4
13	`X-n685-k75.evrp`	477112.4	1791.4	475149.2	479648.1
14	`X-n749-k98.evrp`	440276.3	3127.5	435556.8	443449.7
15	`X-n819-k171.evrp`	600816.5	4980.3	592333.4	605544.4
16	`X-n916-k207.evrp`	753955.5	1834.0	751370.8	756099.5
17	`X-n1001-k43.evrp`	637340.5	2361.7	634329.7	640461.2

runs. All these values are calculated in the output text files generated by the provided source code.

5 Conclusion

In this report we have proposed a set of 17 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.

[3] M. Garey and D. Johnson, *Computer and intractability: A guide to the theory of NP-completeness*. San Francisco: Freeman, 1979.

[4] 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.

[5] A. Froger, J. E. Mendoza, O. Jabali, and G. Laporte, "Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions," *Computers & Operations Research*, vol. 104, pp. 256–294, 2019.

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

[7] 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.