Comparison of different methods to solve Electric Vehicle Routing Problem

Huynh Do Anh Vu – 18521665@gm.uit.edu.vn, Pham Tien Trung – 18521558@gm.uit.edu.vn, Nguyen Kieu Vinh – 18521653@gm.uit.edu.vn, Nguyen Anh Tuan – 18521600@gm.uit.edu.vn

Abstract— **Our research studies the capacitated electric vehicle routing problem (EVRP), is extended by the vehicle routing problem (VRP) and takes into account specific characteristics of electric vehicles. We formulate the mathematical models of the EVRP and compare different method to solve Electric Vehicle Routing Problem. The objective of the EVRP is to minimize the total distance traveled by an electric vehicle (EV). Our research conducts a numerical experiment and sensitivity analysis. The results of the numerical experiment show that these algorithms are capable of obtaining good EVRP solutions within a reasonable amount of time, and the sensitivity analysis demonstrates that the total distance is dependent on the number of customers, the vehicle driving range, battery charge level limits and maximum carrying capacity. Finally, method use MST as the base for Initial Construction method, combine with Randomized VND local search and AFS reallocation give better results than the other methods in the most of benchmarks.**

I. Introduction

Transportation is the main contributor for CO2 waste, which causes global warming, pollution and climate changes. Because of this, many logistic companies such as FedEx, UPS, DHL and TNT are investing in a way to reduce the CO2 waste as a part of their daily operations. This makes the importance of electric vehicles (EVs) increasing by the recent years. The development of electric vehicles has progressed massively, and many of them have been put into use. Although the electric vehicles have many advantages over the traditional vehicles, such as independence of fossil fuels, etc…, it still has one big disadvantage and that is the limited battery's capacity. This gives rise to the demand of new route planning algorithms, which take into account the special properties of EVs. This project is motived by the IEEE WCCI-2020 Competition on Evolutionary Computation for the Electric Vehicle Routing Problem (EVRP) [1]. In this paper, we will present the problem definition in Section II, all the related work in for this problem in Section III. The method we used to solve this problem will be described in Section IV. Section V discusses the experimental results. And finally, Section VI presents conclusions and future direction.

II. Problem Definition

The content of this section is mostly taken over from [1].

The EVRP is a challenging NP-hard combinatorial optimization problem as it is an extension of the ordinary shortest path problem incorporating additional constraints. 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. The objective is to minimize the total distance traveled, while each customer is visited exactly once. And for each route in EV's solution, the total demand of customers does not exceed the maximum EV's maximum carrying capacity and the total energy consumptions does not exceed the total maximum battery's level. All EVs begin and end at the depot, EVs always leave the charging station fully charged (or fully charged and loaded if it goes from depot), and the charging station, including the depot, can be visited multiple time. The charging station will be referred to as AFS for the rest of this paper.

The EVRP can be mathematically formulated as follows:

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

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

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

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

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

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

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

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

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

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

where $V = \{0 \cup I \cup F'\}$ is a set of nodes and $A = \{(i,j) | i,j \in V, i \neq j\}$ is a set of arcs in the fully connected weighted graph $G = (V, A)$. Set I denote the set of customers, set F' denotes set of β_i node copies of each charging station $i \in F$ (i.e., $|F'| = \sum_{i \in F} \beta_i$) and 0 denotes the central depot. Then, x_{ij} is a binary decision variable corresponding to usage of the arc from node $i \in V$ to node $j \in V$ and d_{ij} is the weight of this arc. Variables u_i and u_j denote, respectively, the remaining carrying capacity and remaining battery charge level of an EV on its arrival at node $i \in V$. Finally, the constant h denotes the consumption rate of the EVs, C denotes their maximal carrying capacity, Q the maximal battery charge level, and b_i the demand of each customer $i \in I$.

III. Related Work

According to the [1], the EVRP variant was first solved in [2]. As far as we are concerned, only three papers are dealing with this problem, and for each one of them, the exact problem formulation slightly varies. The first one is [3], which limits the maximum number of routes in addition to the previously introduced constraints. It presents the solution method based on the ant colony system (ACS) algorithm. Initially, a Travelling Salesman Problem (TSP) route visiting all customers is constructed. This route is then turned into a valid EVRP route by a newly proposed fixing method. Then, the ACS modifies the underlying TSP route according to the solution fitness score, and the whole process repeats.

The second one is [4], which also considers the maximum total delivery time on top of the previously introduced constraints. The initial valid EVRP tour is constructed from the nearest neighbor based TSP tour visiting all customers. The solution is then improved in a local search phase while using four different neighborhood operators (swap, insert, insert AFS, and delete AFS). The proposed algorithm uses a simulated annealing mechanism to allow for accepting non-improving moves with a certain probability, thus preventing premature convergence to a local optimum.

Finally, [5] proposes a novel construction method and a metaheuristic algorithm consisting of a local search and a metaheuristic algorithm. The local search combines the Variable Neighborhood Search (VNS) with Randomized Variable Neighborhood Descent (RVND). The operators used in the local search are 2-opt, 2-string, and AFS reallocation. Various other approaches were successfully applied to the numerous variants of the VRP, and many of these can be adapted to the EVRP formulation. For example, a survey [6] focused only on the variants of EVRP.

IV. Methods

NP-hard is impossible to solve in the larger instances in a reasonable time. In order to tackle this problem, we deployed a metaheuristic approach with polynomial time complexity. The EVRP is a constrained variant of the VRP. In those case, an efficient strategy is to select a metaheuristic that has good performance on the original VRP problem and can be easily adapted with respect to the additional constraints of the EVRP. These two criteria are responsed by the metaheuristics performing neighborhood-oriented, so we choose Randomized Variable Neighborhood Descent (RVND), Variable Neighborhood Descent (VND) and Variable Neighborhood Search (VNS) to solve this problem [13].

Local search is an integral part of these metaheuristics, it allows systematic search in a neighborhood of a current solution using local search operators [11]. Local search operators are described in Section IV-B.

Another problem is designing a method for the construction of a valid initial solution. Construction methods are described in Section IV-C.

For formal components description, we define an EVRP tour *T* as a sequence of nodes $T = \{v_0, v_1, ..., v_{n-1}\}$, where v_i is a customer, a depot or an AFS and *n* is the length of the tour *T*. Then, let e_{ij} be an edge from node v_i to node v_j and $w_{i,j}$ be its weight.

A. *Metaheuristic*

1) (Randomized) Variable Neighborhood Descent - (R)VND: VND is a simple metaheuristic used as a local search method in a more complex metaheuristic VNS (Section IV-A2). It include a deterministic variant (VND) and a stochastic one (RVND). The input is a valid EVRP tour *T*, a sequence of local search operators *N*, and a maximum number of fitness evaluations $evals_{max}$ [5]. The operators passed in *N* are described in Section IV-B.

The VND algorithm requires an input as an initial vaild solution and will output a vaild EVRP tour that has fitness equal or better than the initial solution. At each iteration, a local search operator with the first improvement strategy (excluding AFS reallocation) is applied to a solution. The search continue in the next neighborhood until an improvement is found.

Both of VND and RVND perform the local search (according to the best-improvement scenario) sequentially following the operator's order in *N*. In the case of the RVND, the order of the operators is randomly shuffled first, whereas in the VND, the order remains fixed. Each time an improvement is made, the local search is restarted. The VND then starts again from the first operator in *N*, while the RVND randomly reshuffles all operators first. The algorithm terminates either when no improvement is achieved in any of the operators or when a stop condition is met [5].

2) Variable Neighborhood Search (VNS): VNS is a metaheuristic method which is commonly used to find sub-optimal solutions of optimization problems such as the VRP. It systematically changes the searched neighborhood in two phases: an exhaustive local search to reach a local optimum and a perturbation phase, which uses to get out of the corresponding valley.

First, an initial solution is constructed using one of the construction methods described in Section IV-C. Then, the following process repeats. The best known solution *T** is modified by a randomized perturbation, which results in a possibly non-improving current solution *T*. The current solution *T* then passes through a systematic local search, which uses the operators described in Section IV-B. These operators in the local search phase are applied according to the VND or RVND metaheuristics. If the cost of the current solution T is better than the cost of *T**, *T** replaces *T* as the new best solution. The search process terminates when a stop condition is met. The EVRP competition defines it as a maximal number of fitness function evaluations. When this number is reached within the local search, it terminates and sets the stop flag to true, thus terminating the VNS loop.

The perturbation operator is intended to move the search process out of the reach of the local search operators while keeping most of the properties of the current best solution *T**. For this purpose, the Double-Bridge perturbation is used. The Double-Bridge splits the best known solution *T** into $p + 1$ subroutes, with p is randomly selected indices. These subroutes are randomly shuffled, inverted, and reconnected, producing a possibly invalid tour. This tour is then repaired if necessary by the SSF construction described in IV-C and passed to the local search as a valid tour *T*.

B. Local Search

Local search is one of few general approaches to combinatorial optimization problems. The basic idea of local search is that high-quality solutions of an optimization problem can be found by iteratively improving a solution using small modifications, called moves. A local search operator specifies a move type and generates a neighborhood from the current solution. Given the solution s, the neighborhood of a local search operator is the set of solutions N(s) that can be reached from s by applying a single move of that type. After generating the neighborhood of the current solution, the neigborhood is evaluated, and local search uses a move strategy to select at most one solution from the neighborhood N(s) to become the next current solution [11]. The move strategy we used is called steepest descent, it selects the solution from the neighborhood with the best objective function value, provided that this solution is better than the current solution. If there is no better solution in the neighborhood than the current solution then the local optimum has been reached [13].

In general, local search operators for the EVRP can be distinguished between operators for intra-route optimization and operators for inter-route optimization. These two operator types reflect the two tasks that one has to solve in a EVRP: The allocation of customers to routes (inter-route optimization), and the optimization of each route in itself (intra-route optimization) [11]. These two tasks do not necessarily have to be solved in this order, nor sequentially.

The cost update functions δ provided are expressed as a difference between the sum of removed edges weights and the sum of newly added edges weights. Thus, a positive value of the cost update corresponds to an improvement in fitness and vice versa. Only first improvement mode was tested in the local search. In the first improvement, the first improving tour found is accepted, and the search in the current neighborhood is terminated. In both cases, only valid tours are accepted.

1) 2-opt:

The idea of 2-opt is to remove two edges from the considered route and replace them by two new edges, such that a new route is formed [12]. It takes a pair of indices i, j and a tour T as an input and returns a modified tour *T'*, where the sequence of nodes from i to j index is reversed. It must hold, that $i < j, i \geq 1$ and $j < n$.

The cost update function δ_{2-opt} can be evaluated as:

$$\delta_{2-opt} = w_{i-1,i} + w_{j,j+1} - w_{i-1,j} - w_{i,j+1} \tag{11}$$

where the indices are expressed with respect to the tour T.

2) Or-opt:

Our Or-opt is a special case of 2-opt in which two adjacent nodes from two different location will be swapped in the tour. It takes a pair of indices i, j and a tour T as an input and returns a modified tour *T'*, where the adjacent node $(i, i+1)$ and $(j, j+1)$ are swapped. It must hold, that $i + 1 < j, i \geq 1$ and $j < n - 2$.

The cost update function $\delta_{\text{or-opt}}$ can be evaluated as:

$$\delta_{or-opt} = w_{i-1,i} + w_{i+1,i+2} + w_{j-1,j} + w_{j+1,j+2} \\ - w_{i-1,j} - w_{j+1,i+2} - w_{j-1,i} - w_{i+1,j+2} \tag{12}$$

3) Exchange:

The exchange operator behaves similar to the Or-opt operator, but only move one node instead of moving two adjacent nodes. It takes a pair of indices i, j and a tour T as an input and returns a modified tour *T'*, where the node i and j are swapped. It must hold, that $i < j, i \geq 1$ and $j < n - 1$.

The cost update function δ_{exchange} can be evaluated as:

$$\delta_{exchange} = w_{i-1,i} + w_{i,i+1} + w_{j-1,j} + w_{j,j+1} \\ - w_{i-1,j} - w_{j,i+1} - w_{j-1,i} - w_{i,j+1} \tag{13}$$

4) Relocate:

Changing the order of nodes can produce different solutions. Relocate is the process where a selected node is moved from its current position in the tour to another position. Hence, the position of the selected node is relocated. Each relocation of a node produces one outcome [12]. It takes a pair of indices i, j and a tour T as an input and returns a modified tour *T'*, where the node i is removed and inserted to index j. It must hold, that $i < j, i \geq 1$ and $j < n$.

The cost update function δ_{relocate} can be evaluated as:

$$\delta_{relocate} = w_{i-1,i} + w_{i,i+1} + w_{j-1,j} \\ - w_{i-1,i+1} - w_{j-1,i} - w_{i,j} \tag{14}$$

5) AFS reallocation:

This operator will rearrange and remove the duplicate nodes in the valid EVRP tour. The duplicate nodes are the nodes that are continuously identical in the tour. First all the AFS and duplicate node in the tour are removed. Then the modified tour will pass through SSF (see Section IV-C2) to produce a valid EVRP tour. This is the only operator that doesn't have a cost function.

C. *Initial constructions:*

The process to create a vaild EVRP tour includes two phase: initial construction and repair procedure [5]. In the first phase, a TSP feasible solution is created and go through all customer, disregarding the capacity and battery's level cosntraints, starting from and ending with depot. This phase we will use three algorithm to generate the solution: Nearest Neighbour (NN), Clarke-Wright Savings (CWS) and Minimum Spanning Tree (MST). The second phase we will use our developed method to fix the TSP tour into a vaild EVRP tour. Several examples of generated EVRP tours are presented in Figure 1.

1) *Method to construct Initial TSP feasible solution:*

a) *Nearest Neighbour (NN):* The ideal behind this algorithm is rather simple. Starting from depot, the algorithm will choose the next closest customer that has not been visited. Repeat this process until all customers have been visited. After that we just need to add a depot at the end [8].

b) *Clarke-Wright Savings (CWS):* This is the most widely known heuristic algorithm for travelling salesman problem or classical vehicle routing problem. The concept of this algorithm is to determine the routes by the savings list in which values are sorted from largest to smallest [9].

c) *Minimum Spanning Tree (MST):* The concept of this algorithm is to create a graph such that it connects all the nodes (customers and depot), without any circle and with the minimum sum of edge weight [10].

2) Separate Sequential Fixing (SSF): After the initial construction has been created using one of the methods above, the TSP feasible solution then will be repaired to create a valid EVRP solution. SSF splits into two phases. In the first phase, the load constraint is checked sequentially and whenever the next customer's demand is not satisfied, a depot is inserted before them. This will ensure the load constraint is always satisfied before moving to the next phase. In the second phase, SSF makes sure that the current node and the next node (it can either be the depot, customer or AFS) can reach the nearest AFS of its own. If the next node can't be reached, then the former one will ensure that there's always a AFS nearby to add. But if the next node can't reach its nearest AFS, then it will be stuck in an infinite cycle of going back and forth. Thus, the latter one will break this infinite cycle by adding the next node's nearest AFS to the tour as next node. After passing through two phases, the final route will be a valid EVRP tour.

V. Experiments

A. *Benchmark Settings:*

A benchmark of 17 EVRP instances was provided in this competition. These instances are split into two set small (E) and large (X). Each instances contain exactly one depot, while the number of AFSs varies. Because of the time constraint on the project, we only perform experiment on the E instances. The detailed description for the problems is found in [1].

Because the implemented metaheuristic is stochastic, according to the rules of the competition, we must run 20 independent runs with random seeds from 1 to 20 for each run. The stop condition for each run is defined by the maximum number of evaluations with respect to problem size as $evals_{max} = 25000n$ evaluations, where n is the problem size.

In this section, we perform the experiment to see which combination of setup give the best fitness value to the problem's instances. The fitness values will be the mean value of the first 20 run. Benchmark values of tour fitness were provided in each instances files, however, these values are not guaranteed to be optimal. In fact, in some of the instances, some of our method outperform the provided values.

The setup can be broken down to three parts: *Initial Construction method, Randomized or Sequential Local Search* and finally *Use or Not use AFS reallocation* as operator on Local Search.

B. *Results and Discussion:*

Table I shows a comparison of generated EVRP tour's fitness values under different setup, averaged over the first 20 run for each problem's instances. All 12 setup generate a valid solution for all instances.

In the Table I, we see that over all the setups use AFS reallocation as operator have fitness value higher than the setups that don't use. This happened because when the new solution is found, it has already passed through all local search operators and possibly creates a path that is valid and better than the current solution, but if the current path has redundant AFS then the new path will also have those AFS as there is currently no operator to remove them. That is why adding AFS reallocation as an operator will help increase the solution's quality.

Moreover, the setups that use Randomized variant of VND have the better fitness value than the Sequential variant. We suspect that this is because of the first improvement strategy in local search operator. By using this strategy, only the first improvement in cost will be accepted. This helps reduce the complexity of the search and if given enough time and resource, it will converge to local minima. However, the resource we had is limited so using Sequential variant of VND is not very efficient. Instead, using a random operator for every local search call is a better choice because by using random operator, if we are lucky, then we will converge much faster than Sequential one.

And finally, using three method NN, CWS and MST as our base for comparison, it seems that the fitness value of three methods above seems to differ between each setup and give no improvement in our solution.

It seems that the fitness value depends strongly on second and third path and not depends on the method of initial construction.

As shown in the Table I, the current best setup is setup 9, which use MST as the base for Initial Construction method, combine with Randomized VND local search and AFS reallocation

VI. Conclusion And Future Direction

In this paper, we do various experiments on different setup to determine which setup is the best for the EVRP problem.

As mentioned in Section V-B, the current best setup is setup 9. The construction and local search are paired in the VNS metaheuristic, which repeatedly performs Double Bridge perturbation and RVND until the stop condition is reached. Various setups are also tested and implemented, although all the setup we tested still fall far behind the top result in this problem as given in [7]. However, we still get some interesting results when experimenting will the different setup.

As mentioned in Section IV-A, we only run the E-instances in this problem due to time constraint and we are fully aware that this will not give a full insight on the comparison. In the future, we will perform experiment on the X-instances, implement more repair and initial construction methods for comparison and conduct experiments on more metaheuristic algorithm.

References

[1] Michalis Mavrovouniotis, Charalambos Menelaou, Stelios Timotheou, Christos Panayiotou, Georgios Ellinas, and Marios Polycarpou, *Benchmark Set for the IEEE WCCI-2020 Competition on Evolutionary Computation for the Electric Vehicle Routing Problem*, Tech. report, KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, 2020.

[2] F Goncalves, Cardoso S., and Relvas S., *Optimization of distribution network using electric vehicles: A VRP problem*, Tech. report, University of Lisbon, 2011.

[3] Shuai Zhang, Yuvraj Gajpal, and S. S. Appadoo, *A meta-heuristic for capacitated green vehicle routing problem*, Annals of Operations Research 269 (2018), no. 1-2, 753–771.

[4] Nur Mayke Eka Normasari, Vincent F. Yu, Candra Bachtiyar, and Sukoyo, *A simulated annealing heuristic for the capacitated green vehicle routing problem*, Mathematical Problems in Engineering 2019 (2019).

[5] David Woller, Václav Vávra and Viktor Kozák, *Electric Vehicle Routing Problem*.

[6] Tomislav Erdelic, Tonci Caric, and Eduardo Lalla-Ruiz, *A Survey on the Electric Vehicle Routing Problem: Variants and Solution Approaches*, 2019.

[7] KIOS, *CEC-12 Competition on Electric Vehicle Routing Problem*, July 19-24,2020,Glasgow,Scotland,U.K. https://mavrovouniotis.github.io/EVRPcompetition2020/

[8] Wikipedia, *Nearest neighbour algorithm*, 28 February 2021. https://en.wikipedia.org/wiki/Nearest_neighbour_algorithm

[9] Tantikorn Pichpibul and Ruengsak Kawtummachai, *New Enhancement for Clarke-Wright Savings Algorithm to Optimize the Capacitated Vehicle Routing Problem*, European Journal of Scientific Research ISSN 1450-216X Vol.78 No.1 (2012).

[10] Wikipedia, *Minimum spanning tree*, 6 July 2021. https://en.wikipedia.org/wiki/Minimum_spanning_tree

[11] F. Arnold, K. Sorensena, *Knowledge-guided local search for the Vehicle Routing Problem*, Univ. Antwerp, Departement of Engineering Management, ANT/OR - Operations Research Group, May 2019.

[12] L.Sengupta, R. Mariescu-Istodor, P. Fränti, *Which Local Search Operator Works Best for the Open-Loop TSP?*, Univ. Eastern Finland, 80101 Joensuu, Finland, 23 Sep. 2019.

[13] A. Dhahri, A.Mjirda, K. Zidi, K. Ghedira, *A VNS-based heuristic for solving the vehicle routing problem with time windows and vehicle preventive maintenance constraints,* ICCS 2016.

Huynh Do Anh Vu is a student in University of Information Technology (UIT) since 2018. His major is Computer Science, he is currently at the third year and he is in class KHCL2018.3. He is a graduated high school student from THPT Phu Nhuan, HCM.

Pham Tien Trung is a student in University of Information Technology (UIT) since 2018. His major is Computer Science. He is a class president in class KHCL2018.3 and an executive committee of Computer Science faculty. He is a graduated high school student from THPT No 1 Tu Nghia, Quang Ngai.

Nguyen Kieu Vinh is a student in University of Information Technology (UIT) since 2018. His major is Computer Science, he is currently at the third year and he is in class KHCL2018.3. He is a graduated high school student from THPT Nguyen Trung Truc, Kien Giang.

Nguyen Anh Tuan is a student in University of Information Technology (UIT) since 2018. His major is Computer Science, he is currently at the third year and he is in class KHCL2018.3. He is a graduated high school student from THPT Di Linh, Lam Dong.

No	Setup	E-n22-k4.evrp	E-n23-k3.evrp	E-n30-k3.evrp	E-n33-k4.evrp	E-n51-k5.evrp	E-n76-k7.evrp	E-n101-k8.verp
1	NN + Random + Use AFS	**384.678**	**571.947**	**509.470**	840.237	535.235	705.678	858.524
2	NN + Random + Not use AFS	394.174	581.505	520.905	845.034	539.373	709.157	858.626
3	NN + Sequence + Use AFS	385.406	578.827	520.303	848.324	544.196	720.711	873.871
4	NN + Sequence + Not use AFS	396.022	581.936	520.303	847.767	545.066	720.711	873.871
5	CWS + Random + Use AFS	**384.678**	**571.947**	**509.470**	840.309	534.286	**703.904**	857.602
6	CWS + Random + Not use AFS	396.512	586.619	527.471	845.346	540.241	709.998	859.853
7	CWS + Sequence + Use AFS	384.972	580.829	525.905	846.854	546.686	730.380	885.870
8	CWS + Sequence + Not use AFS	396.713	586.745	527.181	847.203	546.692	730.380	885.870
9	MST + Random + Use AFS	**384.678**	**571.947**	**509.470**	**840.166**	**534.225**	705.857	**856.727**
10	MST + Random + Not use AFS	394.449	586.264	527.659	845.484	540.377	709.783	857.325
11	MST + Sequence + Use AFS	385.112	580.279	524.048	848.172	550.124	728.839	887.394
12	MST + Sequence + Not use AFS	394.857	586.207	527.659	848.970	552.444	728.839	887.394

TABLE I: Result from different setup. Average fitness value over the first 20 run.

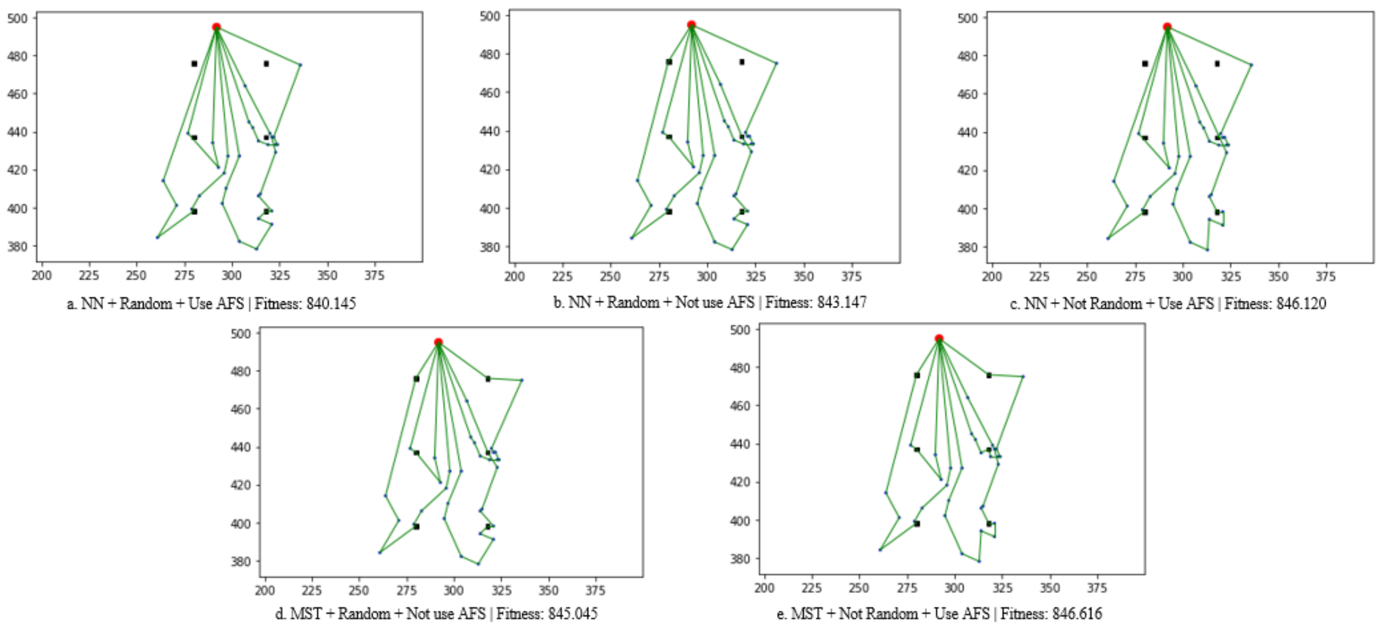

Figure 1: Construction methods, using problem instances E-n33-k4.evrp with random seed 1