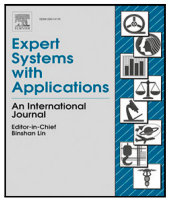
Expert Systems With Applications 252 (2024) 124197

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



A new Hyper-heuristic based on Adaptive Simulated Annealing and Reinforcement Learning for the Capacitated Electric Vehicle Routing Problem

Erick Rodríguez-Esparza [a,*], Antonio D. Masegosa [a,b], Diego Oliva [c], Enrique Onieva [a]

[a] DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain
[b] Ikerbasque, Basque Foundation for Science, Plaza Euskadi, 5, 48009 Bilbao, Spain
[c] Depto. de Ingeniería Electro-Fotónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, 44430, Guadalajara, Jal., Mexico

ARTICLE INFO

Keywords:
Last-mile logistics
Hyper-heuristic
Electric vehicles
Capacitated electric vehicle routing problem
Combinatorial optimization
Reinforcement learning

ABSTRACT

Electric vehicles (EVs) have been adopted in urban areas to reduce environmental pollution and global warming due to the increasing number of freight vehicles. However, there are still deficiencies in routing the trajectories of last-mile logistics that continue to impact social and economic sustainability. For that reason, in this paper, a hyper-heuristic (HH) approach called Hyper-heuristic Adaptive Simulated Annealing with Reinforcement Learning ($HHASA_{RL}$) is proposed. It is composed of a multi-armed bandit method and the self-adaptive Simulated Annealing (SA) metaheuristic algorithm for solving the problem called Capacitated Electric Vehicle Routing Problem (CEVRP). Due to the limited number of charging stations and the travel range of EVs, the EVs must require battery recharging moments in advance and reduce travel times and costs. The implementation of the HH improves multiple minimum best-known solutions and obtains the best mean values for some high-dimensional instances for the proposed benchmark for the IEEE WCCI2020 competition.

1. Introduction

Over the last recent years, there has been a remarkable increase in the use of e-commerce systems around the world, which in turn has had an impact on distribution and last-mile strategies (Castillo et al., 2018; Ignat & Chankov, 2020; Viu-Roig & Alvarez-Palau, 2020). In 2018, a growth rate of 23.3% was reported worldwide (Patella et al., 2021), and these numbers have increased dramatically as a result of the COVID-19 pandemic situation, which has led to huge volumes of packages being delivered daily. Some research reported that some business websites perceived a 74% increase in the e-commerce rate, while 52% of customers avoided in-store purchases (Bhatti et al., 2020; Giuffrida et al., 2022; Singh et al., 2021).

The last-mile refers to the final stage of the delivery journey of a product to reach the end customer. It has been given a greater focus in logistics strategies due to its significant impact on customer satisfaction, delivery time, cost, and convenience (Archetti & Bertazzi, 2021; Vakulenko et al., 2019). As the number of freight vehicles grows in urban areas, last-mile logistics operations have a considerable impact on three different aspects of sustainability: economic (efficiency and delivery costs), social (congestion and health problems), and environmental (CO_2 emissions and noise pollution) (Janjevic et al., 2019). One of the measures that have been taken to address the environmental impact is the adoption of electric vehicles (EVs) since the transportation sector is estimated to be responsible for about 20%–25% of global $C0_2$ emissions (Bosona, 2020; Fafoutellis et al., 2021; He et al., 2018; Yi et al., 2018).

Furthermore, in line with this need to improve the efficiency of last-mile logistics, applying the Vehicle Route Problem (VRP) concept to this field began to gain importance in the research community. Its main objective is to optimize the routes for the transport of goods from one or several warehouses to a set of geographically dispersed clients to increase efficiency and reduce time and costs; in this way, it is intended to combat the social and economic impact (Zirour, 2008). However, the classical models do not provide an answer to the particularities of increasing the use of EVs in last-mile logistics that we mentioned above. The main reason is that they do not consider restrictions in terms of vehicle autonomy and the need to recharge the batteries. Thus, the concept of the Electric Vehicle Problem (EVRP) arises as a variation of the conventional VRP.

The approaches proposed for solving VRP, EVRP, and variants can be classified as: exact algorithms, heuristics, metaheuristics, and hyper-heuristics (HHs) (Asghari & e hashem, 2020; Blocho, 2020). An exact

* Corresponding author.
E-mail addresses: erick.rodriguez@deusto.es (E. Rodríguez-Esparza), ad.masegosa@deusto.es (A.D. Masegosa), diego.oliva@cucei.udg.mx (D. Oliva), enrique.onieva@deusto.es (E. Onieva).

https://doi.org/10.1016/j.eswa.2024.124197
Received 28 September 2023; Received in revised form 7 May 2024; Accepted 8 May 2024
Available online 20 May 2024

algorithm always finds the optimal solution. Still, it has the disadvantage that it can only tackle problems of relatively small size due to its high computational time requirements (Purkayastha et al., 2020). Metaheuristic algorithms are comprehensive techniques that provide a general structure and strategic criteria for developing a heuristic method to solve the problem (Fausto et al., 2020; Morales-Castañeda et al., 2020; Oliva et al., 2020). Heuristics and metaheuristics are good alternatives for solving VRP and its variations, including EVRP. Nevertheless, deciding when to apply a specific method or operator (Osaba et al., 2020; Rodríguez-Esparza et al., 2024) can be challenging.

Moreover, there are still difficulties in using the current algorithms due to the existence of multiple parameters and high sensitivity in their configuration (Swiercz, 2017). Therefore, choosing a proper search mechanism is crucial since it allows obtaining optimal results with high accuracy. In this sense, the use of HHs is becoming increasingly popular. HHs are considered high-level automatic search methods that work by combining, generating, and selecting low-level heuristics to solve computationally complex problems (Burke et al., 2013).

HH algorithms generally involve two main components: named selection and move acceptance. The selection mechanism helps to identify which element from the pool of low-level heuristics should be applied at each stage of the solving process (Scoczynski et al., 2021b). The move acceptance mechanism decides whether to accept or discard the generated solution (Turky et al., 2018). HHs are powerful tools, however, the design of heuristic selection is complex and time-consuming; because it is usually done by trial and error that depends on the problem (Choong et al., 2018; Drake et al., 2020; Zhang et al., 2021).

To address these challenges, leveraging machine learning mechanisms, including Reinforcement Learning (RL) strategies, offers a promising avenue for creating intelligent and self-adaptive algorithms (de Santiago Júnior et al., 2020; Largo et al., 2020; Wang & Tang, 2021). RL methods are learning strategies to find the best actions to apply for a particular state or observation of/from an environment. The agent chooses actions sequentially in discrete time steps from a set of available actions and receives a reward that varies depending on the utility of the action taken. The actions taken affect or change the state of the environment of the agent.

Based on the ideas outlined above, in this paper, we propose a HH to address a variant of the EVRP efficiently, called the Capacitated Electric Vehicle Routing Problem (CEVRP). The main components of the proposed algorithm called Hyper-heuristic Adaptive Simulated Annealing and Reinforcement Learning ($HHASA_{RL}$), are a RL algorithm as a selection mechanism and the Metropolis criterion of the well-known Simulated Annealing (SA) metaheuristic algorithm as the movement acceptance mechanism. The selection problem among the pool of heuristics is treated as a multi-armed bandit problem (Slivkins, 2019). This stochastic problem is used for the dilemma of exploration and exploitation of the algorithm. It is assigned a fixed set of options or actions, and the agent selects one within that set to maximize the long-term cumulative reward.

A comparison of three of the most commonly used algorithms for dealing with multi-armed bandit problems is presented in this paper. The RL algorithms used for this comparison are Epsilon Greedy (ϵ-G), Thompson Sampling (TS), and Upper Confidence Bound 1 ($UCB1$) to identify which of them most efficiently selects the low-level heuristic for the CEVRP. These techniques guide and control the local search of the SA by choosing the heuristic that is best applied during the iterative process of the algorithm to optimize the long-term performance by improving the quality of the solutions.

The most relevant contributions of this proposal can be summarized as follows:

- A methodology for solving high-dimensional electric vehicle routing problems.
- A hyper-heuristic based on hybridization of self-adaptive simulated annealing and reinforcement learning treating it as a multi-armed bandit problem.
- State-of-the-art results for the CEVRP benchmark proposed for the IEEE WCCI2020 competition.

To validate this proposal's performance and competitiveness, we have used instances of the CEVRP from the IEEE WCCI 2020 competition. This benchmark contains 17 instances of short and long problems, ranging from 21 to 1000 customers. Statistical analysis and non-parametric tests were conducted to validate the proposed results of the algorithm in comparison with the rest of the approaches.

The remainder of this article is organized as follows: Section 2 provides a formal description of the CEVRP. Then, Section 3 shows the related works to solve CEVRP. Section 4 gives background information about important concepts used for the proposed HH. Section 5 details the proposed $HHASA_{RL}$ algorithm. Afterward, Section 6 presents the experimental framework. In Section 7, the experiments and results are shown. Finally, in Section 8 the conclusions and further work are included.

2. The capacitated electric vehicle routing problem

The CEVRP is a combinatorial optimization problem classified as $\mathcal{NP}$-hard. It is a variation of the traditional VRP, incorporating capacity constraints. In CEVRP, a fleet of EVs with specific load and battery capacity aims to find optimal routes, minimizing the total travel distance to fulfill the demands of a set of customers while adhering to various constraints. Unlike VRP, CEVRP accounts for specific EV characteristics such as vehicle autonomy and the need for battery recharging (Erdelić & Carić, 2019; Mavrovouniotis et al., 2018).

The CEVRP is defined on a complete, undirected graph $G(V, A)$, where $V = \{D \cup C \cup S\}$ and $A = \{(i, j) \mid i, j \in V, i \neq j\}$. Here, V represents nodes, comprising a set C of n_c customers, a set S of n_s external charging stations, and a central depot denoted by D. The set of arcs connecting the nodes is denoted by A. Each arc has a non-negative value d_{ij} representing the distance between nodes i and j. When the EV travels along an arc (i, j), it consumes energy $e_{i,j} = h \cdot d_{ij}$, where the parameter h is the energy consumption rate.

Each customer i has a specific delivery demand q_i. All EVs are identical, with a maximum load capacity (Max_C) and a maximum battery capacity (Max_Q). These parameters should not be exceeded for each EV. The EVs start (fully loaded and charged) and end at the depot; it is important to mention that each of the vehicles can visit the charging stations several times, but all customers must be visited exactly once.

The CEVRP mathematical model is formulated as follows (Mavrovouniotis et al., 2020a):

$$min \quad f(\mathbf{x}) = \sum_{i \in V, j \in V, i \neq j} d_{ij} \cdot x_{ij}, \tag{1a}$$

$$s.t. \quad \sum_{j \in V, i \neq j} x_{ij} = 1, \qquad \forall i \in C, \tag{1b}$$

$$\sum_{j \in V, i \neq j} x_{ij} \leq 1, \qquad \forall i \in S, \tag{1c}$$

$$\sum_{j \in V, i \neq j} x_{ij} - \sum_{j \in V, i \neq j} x_{ji} = 0, \qquad \forall i \in V, \tag{1d}$$

$$u_j \leq u_i - c_i \cdot x_{ij} + Max_C \cdot (1 - x_{ij}), \qquad \forall i \in V, \forall j \in V, i \neq j, \tag{1e}$$

$$0 \leq u_i \leq Max_C, \qquad \forall i \in V, \tag{1f}$$

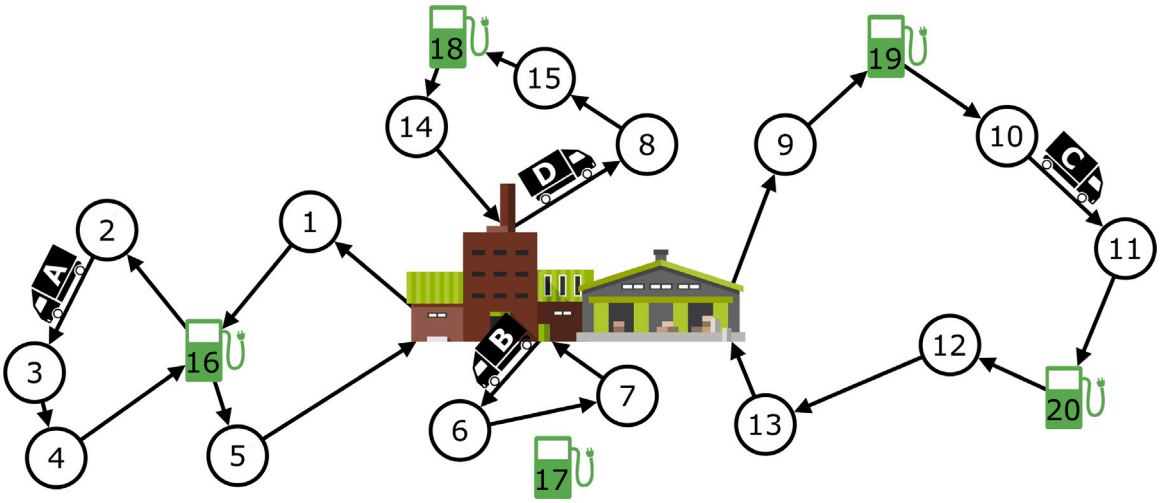


Fig. 1. Example of CEVRP with four routes. (A) stops twice in the same charging station, (B) does not stop in any station, (C) stops in two different stations and (D) stops in a single station.

$$y_j \le y_i - hd_{ij} \cdot x_{ij} + Max_Q \cdot (1 - x_{ij}), \quad \forall i \in I, \forall j \in V, i \ne j, \tag{1g}$$

$$y_j \le Max_Q - hd_{ij} \cdot x_{ij}, \forall i \in S \cup \{0\}, \quad \forall j \in V, i \ne j, \tag{1h}$$

$$0 \le y_i \le Max_Q, \quad \forall i \in V, \tag{1i}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V, \forall j \in V, i \ne j, \tag{1j}$$

where Eq. (1a) defines the CEVRP objective function, aiming to minimize the total travel distance of all EVs. Constraint in Eq. (1b) ensures each customer is served only once, while Eq. (1c) allows charging stations to be visited multiple times. Eq. (1d) establishes flow conservation by guaranteeing that at each node, the number of incoming arcs is equal to the number of outgoing arcs. The Eqs. (1e) and (1f) are the capacity constraints that guarantee that the load of an EV is non-negative upon arrival at any node, including the deposit. And the energy constraints of Eqs. (1g), (1h) and (1i) ensure that the battery charge level never drops below 0. Finally, Eq. (1j) defines the set of binary decision variables (x_{ij}); if the arc (i, j) is traveled by an EV, x_{ij} is equal to one and zero otherwise.

The variables u_i and y_i, respectively, represent the remaining charge capacity and the remaining energy level of the EV when it reaches the node $i \in V$. Despite explicit constraints, all EVs must depart and return from the depot.

2.1. Route representation and charging station encoding

Fig. 1 illustrates various scenarios of CEVRP routes for four EVs, denoted by A, B, C, and D. These scenarios showcase different cases that may occur in the routes, highlighting the versatility of the proposed algorithm.

In the first case, shown in route A, the EV visits the same charging station twice, which may happen in certain routing situations. In contrast, route B demonstrates that the EV will not run out of battery, avoiding the need to pass through any charging station. On the other hand, route C illustrates a long journey where the EV needs to pass through two different charging stations. This scenario showcases the ability of the algorithm to handle complex routing requirements. Lastly, route D shows that the EV only passes through the charging station once before returning to the depot, demonstrating flexibility in route planning.

The algorithm encodes each route as a sequence of numbers. The depot is represented by the number 0, and numbers from 1 to $n_c + n_s$ uniquely represent both route customers and charging stations. The EV routes are separated by 0s, indicating that each route must start and end at the depot. To add recharging stations, the encoding scheme utilizes the number −1. In the encoded sequence, the −1 is replaced by the index of the nearest charging station between the two clients.

Consider the example of EV A shown in Fig. 1. The corresponding encoding for EV A would be [0, 1, 16, 2, 3, 4, 16, 5, 0]. In this sequence, the underlined numbers represent the charging stations inserted into the route. To elaborate further, let us examine the four routes presented in Fig. 1:

- Route A: [0, 1, 16, 2, 3, 4, 16, 5, 0]
- Route B: [0, 6, 7, 0]
- Route C: [0, 8, 15, 18, 14, 0]
- Route D: [0, 9, 19, 10, 11, 20, 12, 13, 0]

This encoding scheme ensures a concise representation of CEVRP routes, facilitating the comprehension and implementation of the algorithm.

3. Related work

In this section, some significant works proposed in the state-of-the-art to solve CEVRP are described. This problem can be seen as a variant of the green vehicle routing problem first proposed in 2012 by Erdogan and Miller-Hooks (Erdoğan & Miller-Hooks, 2012). The authors used a modified Clarke and Wright savings heuristic and a density-based clustering algorithm to find initial solutions, followed by an optimization phase. The green vehicle routing problem generally refers to routing alternative fuel vehicles, including EVs, which present the characteristic of having a limited travel range. Therefore, they require planning for battery recharging during the delivery route.

Over recent years, the scientific literature on EVRP has grown uninterrupted. In 2015, Pelletier et al. presented a detailed study of the variants of the types of EVRP (Pelletier et al., 2016). Later in 2019, Erdelić and Carić conducted a survey in which they classified articles according to the composition of the fleet, the terms of the objective function, the presence of multiple recharging technologies, and limitations such as capacities of the EVs and time windows for the customers. In addition, their work reviewed the exact, heuristic, metaheuristic, and hybrid approaches applied to solve different variants of EVRP (Erdelić & Carić, 2019). As can be seen from the works

mentioned above on the state-of-the-art, there are different strategies to solve problems with variations in EVs. As is the case in the work of Scheinder et al. in which they extended the model to EVRP by adding time windows for the customers and solved it through hybridization between the VNS algorithm with a TS heuristic (Schneider et al., 2014). Their proposal was tested in 56 instances based in the benchmark proposed by Solomon (1987) with up to 100 clients.

Keskin and Çatay used the same instances of the Solomon benchmark (Keskin & Çatay, 2016). They considered the cases with a partial recharge of the battery when stopping at a recharging station and different objectives. They proposed two works using an ALNS approach by introducing new route modification heuristics to add/remove clients and stations (Keskin & Çatay, 2018). Montoya et al. proposed an automated repair strategy that inserts stations into routes to ensure route viability and a modified multi-space sampling heuristic, which was tested on 52 cases with up to 500 clients (Montoya et al., 2016). One year later, Montoya et al. formulated a recharging of battery process as a nonlinear function and presented a hybrid metaheuristic combining an iterated local search and a heuristic concentration to solve the new variation of the problem called EVRPNL (Montoya et al., 2017). This proposal was tested on 120 problems, with customers ranging from 10 to 320. Recently, Mao et al. investigated a new variation of EVRP, named EVRPTW&MC, in which they added decisions on multiple recharging options, which are partial recharging and battery swapping (Mao et al., 2020). An improved ACO algorithm was proposed that is combined with an insertion heuristic and an improved local search. The ACO was compared and validated using 56 experimental instances used in other works in the literature (Schneider et al., 2014).

On the other hand, there are also some proposals in the state-of-the-art where the researchers use RL techniques to solve the EVRP and its variations due to the advantages mentioned in Section 1. This is the case of Shi et al. who introduced a new off-policy RL framework with decentralized learning and centralized decision-making processes used to solve the EVRP in the ride-hailing services (Shi et al., 2019). In 2020, Lin et al. proposed a deep RL framework for solving the EVRPTW that used an attention model with a pointer network and a graph embedding layer to parameterize a stochastic policy. The REINFORCE gradient estimator trains the model, which has a greedy roll-out baseline. Their proposal was tested in seven small problems with client numbers ranging from 5 to 100 (Lin et al., 2020). Also, Zhao et al. hybridized a deep RL model composed of an actor, an adaptive critic, and a routing simulator with a local search strategy to increase solution quality even more. This proposal was tested on three datasets with up to 100 clients to solve the VRP and VRP with time windows (Zhao et al., 2020). Recently, Bogyrbayeva et al. have developed an RL framework-based method for free-floating electric vehicle sharing systems, in which a central controller determines the routing strategies of a fleet of various shuttles. Using a policy gradient method, they train a recurrent neural network and compare the outcomes to heuristic solutions. For this investigation, they employed three problems with up to 100 customers (Bogyrbayeva et al., 2021).

From the above, it can be seen that there are efficient and novel approaches in the state-of-the-art of EVRP. However, despite the effectiveness of these works, they still have the disadvantage of not being stable and robust when increasing the problem's dimensionality. Generally, most of these methodologies are only used for EV routing with a maximum of 100 to 320 clients. Therefore, in this article, we propose an intelligent and self-adaptive algorithm called HHASA$_{RL}$ to solve CEVRP efficiently, obtaining optimal results at high-dimensional problems.

4. Background

This section provides background information on the essential building blocks for the HH proposed in this paper. First, the HHs are detailed by discussing their importance, classification, and the most commonly used techniques. Subsequently, the SA algorithm, as well as the Metropolis criterion, are presented. Then, algorithms for handling multi-armed bandit problems are detailed, followed by an overview of commonly used improvement heuristics.

4.1. Hyper-heuristics

HHs represent a class of high-level automated search techniques designed to enhance the generality and robustness of search methods for solving more complex problems. These algorithms explore a search space of low-level heuristics that can be neighborhood or movement operators, heuristic, or metaheuristic algorithms. The two main categories of HHs are heuristic generators and heuristic selectors. Heuristic generators create new heuristics from components of existing ones, while heuristic selectors choose a heuristic from a set. This research focuses on HHs for selection, which control a set of low-level heuristics during an iterative search process.

A generic heuristic selection approach comprises two key components: heuristic selection and move acceptance. The heuristic selection strategy involves choosing the most suitable low-level heuristic from a set of heuristics at a specific point during the search process. Meanwhile, the move acceptance strategy determines whether to accept or reject the solution generated by the previous component.

Among the most straightforward methods are random selection, random gradient, and greedy search. Random selection and random gradient involve selecting a heuristic randomly and iteratively applying it until no further improvement in fitness is observed. On the other hand, greedy search utilizes all perturbative heuristics from the available set and selects the one with the best fitness (Pillay & Qu, 2018).

Some strategies more commonly used by metaheuristic algorithms, such as tournament selection and roulette wheel, have also been used. The roulette wheel strategy associates each heuristic with a probability calculated by dividing its score by the total score. Subsequently, a heuristic is randomly chosen based on these probabilities. Tournament selection involves randomly selecting a set of heuristics of fixed size to participate in several tournaments, and the heuristic achieving the best fitness is ultimately selected (Burke et al., 2013).

RL has been used successfully in assigning scores to each heuristic within the available pool based on their performance during the iterative process. During this process, the system learns heuristics through trial and error, evaluating the status and accumulated rewards of actions (Burke et al., 2013). In the same way, the most commonly used movement acceptance techniques are presented. The simplest strategy is to accept all moves regardless of the quality of the solutions, and another straightforward approach is to accept only moves that improve the fitness of the solution. Local search techniques such as SA, late acceptance hill-climbing, and great deluge have been utilized based on their specific strategies (Pillay & Qu, 2018).

4.2. Simulated annealing

This metaheuristic algorithm inspired by the physical process of annealing solid metals in metallurgy was proposed by Kirkpatrick et al. in 1983 to solve both global and combinatorial optimization problems (Kirkpatrick et al., 1983).

Taking the thermodynamic system as a reference, a candidate solution is generated in each iteration, considering improvement heuristics. The new candidate solution is accepted or rejected according to the Metropolis relation. This acceptance criterion is shown in Eq. (2) and is the key aspect of the SA algorithm to avoid stagnation at local optima. The Metropolis criterion uses the relative quality of the solution and the temperature as a probability to select worse solutions, promoting exploration of the search space (Delahaye et al., 2019).

$$p^k = \begin{cases} \exp\left(\frac{-\Delta}{T}\right), & \text{if} \quad \Delta > 0 \\ 1, & \text{if} \quad \Delta \leq 0 \end{cases} \tag{2}$$

Algorithm 1 Pseudo-code of SA

Inputs: I_{Iter}, α, T_0, M_{Acc}
s ← Create initial solution
$T \leftarrow T_0$
while $acc < M_{Acc}$ **do**
  for (k ← 1 to I_{Iter}) **do**
    $\mathbf{s}'$ ← Create neighbor solution(**s**)
    $\Delta = f(\mathbf{s}') - f(\mathbf{s})$
    if $\Delta \leq 0$ **then**
      $\mathbf{s} \leftarrow \mathbf{s}'$
    else
      if $p^k > rand$ **then**
        $\mathbf{s} \leftarrow \mathbf{s}'$
    $acc \leftarrow acc + 1$
    k ← k+1;
  $T = T \cdot \alpha$
Return: best solution found

The original pseudo-code of the SA is observed in Algorithm 1.

where I_{Iter} specifies the number of iterations for which the local search continues at a particular temperature. At the same time, α is the coefficient controlling the cooling schedule, and T_0 is the initial temperature equal to the current temperature (T) at the beginning of the algorithm. The M_{Acc} represents the maximum number of function accesses allowed (Morales-Castaneda et al., 2019).

4.3. Multi-armed bandit RL methods

The multi-armed bandit is a well-known problem in RL problem whose name originates from a gambler sitting in front of a set of n-slot machines. The objective is to obtain the highest value of the accumulated reward between each spin and using one machine at a time, whether to continue playing with the current machine or to switch to another one (Auer et al., 2002; Gittins et al., 2011). It optimizes its reward by acquiring knowledge (exploration) and optimizing decisions based on that learning (exploitation). Formally expressed and simply, the multi-armed bandit problem of K-arms (actions or heuristics) is defined by $A_{i,n}$, which are random variables from $1 \leq i \leq K$ and $n \geq 1$, where i represents the index of the slot machine. There are many different solutions to tackle the multi-armed bandit problem. However, the most commonly used methods are Epsilon Greedy (ϵ-G), Thompson Sampling (TS), and Upper Confidence Bound 1 ($UCB1$).

4.3.1. Epsilon greedy

The ϵ-G method is a simple method to balance exploration and exploitation when choosing between random exploration and exploitation. As its name suggests, it is considered the greediest algorithm among the other two algorithms presented below. The ϵ value (ranging from 0 to 1) determines the probability of choosing to explore. However, the constant is usually set to 0.1, indicating that it exploits most of the time with a small possibility of exploring (Yang et al., 2021). The pseudo-code of this strategy is presented in Algorithm 2. The vector $\boldsymbol{R}$ keeps track of the accumulated rewards for each action or heuristic up to that moment.

Algorithm 2 Pseudo-code of ϵ-G

Inputs: ϵ, $\boldsymbol{R}$
if ($rand > \epsilon$) **then**
  $heuristic$ ← Select a random action
else
  $heuristic$ ← Select the action with the $argmax(\boldsymbol{R})$
Return: $heuristic$

4.3.2. Thompson Sampling

The TS approach is more based on Bayesian principles and can produce more balanced and efficient results in some cases. It involves constructing a probability distribution, typically a beta distribution, to represent the actual success rate of each action. This distribution is built using information from previously taken actions as training, creating an active exploration with a trial-and-error search of the behavior of the actions in each of the moves (Russo et al., 2017). The algorithm then decides whether to reward or punish each action based on heuristic results, which increases the corresponding value of vector $\boldsymbol{R}$ or vector $\boldsymbol{P}$, respectively. This strategy generates other actions that probably maximize the reward, leading to future performance improvements. The pseudo-code for this approach is presented in Algorithm 3.

The probability of selecting a heuristic is proportional to the number of times the corresponding machine (action) completes successfully compared to failures. Nevertheless, heuristics with a lower success/failure ratio may still be chosen, to enable exploration.

Algorithm 3 Pseudo-code of TS

Inputs: $\boldsymbol{R}$, $\boldsymbol{P}$
for (i ← 1 to $num_{actions}$) **do**
  $\theta_i \leftarrow Beta(R_i + 1, P_i + 1)$ sample from Beta distribution
$heuristic$ ← Select the action with the $argmax(\boldsymbol{\theta})$
Return: $heuristic$

4.3.3. Upper Confidence Bound 1

The $UCB1$ is a decision-making algorithm grounded in the principle of optimism in the face of uncertainty. When uncertain about which action to take, the algorithm adopts an optimistic stance, assuming optimistically that the chosen action is correct. The main idea behind $UCB1$ is to consistently select the heuristic with the highest upper bound, effectively striking a balance between exploration and exploitation, which is one critical characteristic of the algorithm (Umami & Rahmawati, 2021).

Due to its exploration to systematically reduce uncertainty, its exploration decreases over time because it decays exponentially as the number of turns or shots of the machines increases. In other words, the least explored machine gets a boost even if its estimated average is low, especially if the gambler has been playing for a while. This distinctive characteristic allows $UCB1$ to define its exploration–exploitation combinations without depending on any parameters.

The pseudo-code is outlined in Algorithm 4. The vector $\boldsymbol{R}$ illustrates the accumulated rewards for each action, and the vector $\boldsymbol{S}$ indicates the number of times each heuristic has been selected.

Algorithm 4 Pseudo-code of $UCB1$

Inputs: k, $\boldsymbol{R}$, $\boldsymbol{S}$
if ($k \leq num_{actions}$) **then**
  $heuristic \leftarrow k$
else
  for (i ← 1 to $num_{actions}$) **do**
    $\phi_i \leftarrow R_i/S_i + \frac{\sqrt{2 \cdot \log(k)}}{S_i}$
$heuristic$ ← Select the action with the $argmax(\boldsymbol{\phi})$
$S_{heuristic}$ ← Is increased by 1
Return: $heuristic$, $\boldsymbol{S}$

4.4. Heuristics for VRP

Heuristics aim to address problems based on specific domain knowledge, typically yielding suboptimal or sufficiently close to satisfactory solutions. In the research field of VRP, heuristics are categorized into constructive and improvement heuristics. Constructive heuristics are iterative methods employed to generate initial solutions by sequentially

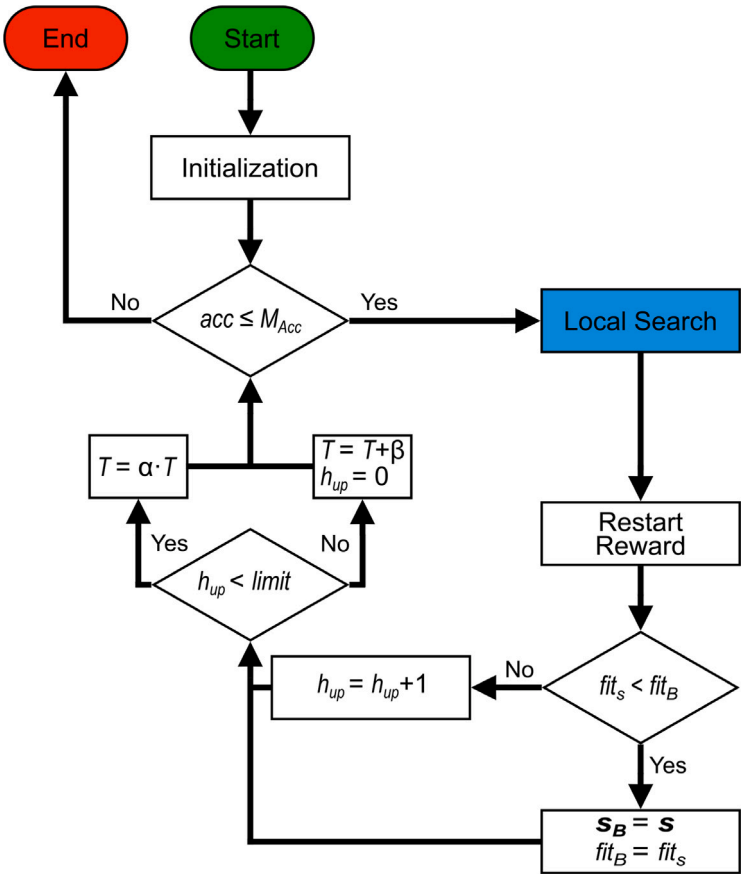


Fig. 2. The general flowchart of HHASA$_{RL}$.

constructing routes and adding elements until a complete solution is formed. These solutions are built greedily and often deviate from the optimal solution significantly (Vidal et al., 2013). On the other hand, improvement heuristics explore the neighborhood of the current solution, searching for a better solution by applying perturbation operators. The local search process concludes when no improved solution is found in the neighborhood, leading to a state referred to as a local optimum. The VRP literature commonly employs the following perturbation heuristics:

- *Swap*: Two nodes, selected randomly, exchange positions, either within the same route or on different routes.
- *Reversion*: Starting from two nodes, the number string is inverted regardless of whether they are not on the same route.
- 2*Opt*: Two randomly chosen arcs are replaced with two new ones, with the option to reverse the direction of the route upon reconnection.
- *Insertion*: Two nodes are selected, and the first node is inserted in the position following the second selected node.

5. Hyper-heuristic adaptive simulated annealing and reinforcement learning

This section offers a comprehensive insight into the HHASA$_{RL}$ algorithm. It begins by detailing the key components constituting the general flowchart of HH. Subsequently, it delves into the local search block, consisting of four significant sub-blocks: *RL Method*, *Generate*, *Repair*, and *Adjust Station*.

To enhance the understanding of the intricacies of heuristic selection in HHASA$_{RL}$, parallels are drawn with the multi-armed bandit problem. In this context, heuristics act as arms, each with its associated rewards. The exploration–exploitation dilemma is resolved by leveraging RL techniques to dynamically assign scores to each heuristic based on its performance during the iterative process.

This adaptation of the multi-armed bandit paradigm facilitates the ability of the algorithm to make informed decisions during the local search. The RL method in the local search block (Section 5.2) is crucial for selecting heuristics that balance exploration and exploitation to obtain an efficient solution.

5.1. General description

The general process of the HHASA$_{RL}$ algorithm is illustrated in Fig. 2. This method inherits the same input parameters as the classical SA algorithm (detailed in Section 4.2), encompassing T, $MAcc$, I_{Iter}, and α. Additionally, parameters $limit$ and variable h_{up} are introduced to regulate the temperature of the proposal. The parameter $limit$ serves as a threshold, indicating the maximum number of iterations that h_{up} can reach without improving the solution. If this condition is met, the T value increases.

The dynamic temperature control, governed by the variable β, plays a pivotal role in maintaining the balance between exploration and exploitation during the local search.

Exploration is encouraged by accepting worse solutions in search of unexplored regions of the search space. Meanwhile, exploitation is maintained during the cooling phase, where solutions are accepted primarily if they are better than the best global solution. This dynamic temperature control mechanism allows the algorithm to adapt to different characteristics of the search landscape, promoting exploration when needed and adjusting to exploitation in promising regions, thereby enhancing the ability of the algorithm to find high-quality solutions in the CEVRP problem space.

The HHASA$_{RL}$ general flowchart consists of four main steps that are outlined below:

1. **Initialization Procedure:** The process starts by initializing the internal variables and generating a feasible random solution (s). This solution is obtained by randomly permuting all customers, organizing them into vehicle routes based on load, and strategically adding stops at the nearest stations to recharge the battery (before emptying the battery).
2. **Iterative Loop with Perturbations:** The generated solution s undergoes a loop where perturbations or local changes are applied in the algorithm until the maximum allowed number of fitness function accesses is reached (while $acc \leq M_{acc}$). The variable acc controls the number of evaluation function accesses used throughout the iterative loop process, ensuring it does not exceed the maximum number.
3. **Local Search Block:** During this stage, I_{Iter} perturbations are executed to update the current solution s and minimize the total distance of the tours. The reward vector is then reset, and the fitness of the best solution generated in the local search, fit_s, is compared with the fitness of the global best solution, fit_B. If the route distance improves, the global solution is updated. Otherwise, h_{up} is increased by 1. The variable h_{up} determines whether the temperature should continue cooling or if it requires reheating. This block is comprehensively detailed in Section 5.2.
4. **Temperature Adjustment:** Depending on the value of h_{up} in relation to $limit$, the temperature is adjusted accordingly. If h_{up} is less than $limit$, indicating a steady improvement in fitness, the temperature continues to decrease. The process starts again with the *Local Search* block, utilizing the new temperature obtained through $T = \alpha \cdot T$. On the other hand, if h_{up} is equal to the $limit$ value, signifying a lack of improvement in the solution, the temperature undergoes reheating. The amount added to the current temperature is determined by the variable β in Eq. (3) to increase the probability of accepting worse solutions to escape from local minima through the Metropolis relation.
   The value of β, influenced by the variable acc representing the current number of accesses to the objective function, is calculated using the linear decrement formula presented in Eq. (3) between the points $(x_{\text{ini}}, y_{\text{ini}})$ and $(x_{\text{end}}, y_{\text{end}})$.

$$m = (y_{end} - y_{ini})/(x_{end} - x_{ini}) \tag{3a}$$

$$\beta = m \cdot \left(\frac{acc}{M_{acc}} \cdot 100 \right) + y_{ini} \tag{3b}$$

   where x represents percentages of accesses to the objective function used, with x_{ini} starting at the beginning of the algorithm, corresponding to a percentage of 0. x_{end} represents the final percentage by which the temperature can be heated if the fitness does not improve in $limit$ local searches.
   On the other hand, the values of y represent the dynamic temperatures added to heat the system in the case of no improvement in fitness. y_{ini} denotes the initial temperature increased for reheating, while y_{end} is the final temperature added. A visualization of the behavior of the β dynamic temperature is provided in Fig. A.6 in Appendix A.

5.2. Local search block

The flowchart depicted in Fig. 3 outlines the local search process in the proposed algorithm. This local search spans I_{Iter} iterations, maintaining a constant temperature throughout the cycle.

1. **RL Method Block:** The *RL Method* block employs a multi-armed bandit RL method (as detailed in Section 4.3) to select a suitable heuristic. This decision relies on the information from the reward vector specific to the ongoing local search. The RL pool consists of eight commonly used perturbation heuristics detailed in Section 4.4: $Swap_{r1}$, $Reversion_{r1}$, $2Opt_{r1}$, $Insertion_{r1}$, $Swap_{r2}$, $Reversion_{r2}$, $2Opt_{r2}$, and $Insertion_{r2}$. The heuristics are divided into two groups of four each. The first four have a subindex $r1$, while the second four have a subindex $r2$. Each of these subindices corresponds to a percentage, determining the closeness range within the customer set between customer c_1 and customer c_2 relative to the total number of customers (n_c) in each problem instance. Consequently, this indicates that the heuristic randomly chooses customer c_2 based on the closeness of the specified percentage interval of its subindex to customer c_1.
2. **Generate Block:** The *Generate* block constructs a new solution s' using the chosen heuristic. The Algorithm 5 presents the pseudo-code for this block. This heuristic operates on customers c_1 and c_2, modifying the solution s. To prevent the repetition of customer c_1, a random vector comprising all customers, called ***cust***, is created. Customer c_1 is assigned and removed from the first element of ***cust*** to facilitate the algorithm process. After all customers in the current vector have been considered, ***cust*** is regenerated with a new random order, ensuring a continuous supply of unique clients.

Algorithm 5 Pseudo-code of *Generate block*

Inputs: s, $heuristic$, n_c, ***cust***
if ∼ isempty(***cust***) **then**
  cust ← permutation(n_c)
$c_1 \leftarrow cust(1)$
Remove c_1 from ***cust***
Select c_2
$s' \leftarrow$ Apply the $heuristic$ to c_1 and c_2
$s' \leftarrow$ Eliminate stations that are next to each other
Return: s', ***cust***

3. **Repair Block:** After generating a new solution, the load capacity constraint for s' is verified. If the constraint is unsatisfied, the *Repair* block is activated to adjust the generated route and ensure that EV loads remain feasible. The pseudo-code is outlined in Algorithm 6.
   The algorithm stores the last customers that exceed the EV load capacity in a vector called ***repair***. Subsequently, the block evaluates other routes to determine the feasibility of adding these customers to each route, ensuring that the EV load limit is not exceeded.
   During load repairs, two possibilities arise. Firstly, if there is enough capacity in other EVs to accommodate these loads, customers are added at the route position next to the customer with the minimum distance. On the other hand, if there is no available space in other EVs for additional loads, the new solution s' cannot be repaired. In this scenario, the variable k is incremented, and the process starts anew, beginning with heuristic selection in the *RL Method* block. This decision is pivotal in avoiding the unregulated creation of new routes that could lead to suboptimal convergence.

Algorithm 6 Pseudo-code of *Repair* block

Inputs: s'
for (v ← 1 to num_{routes}) **do**
  q_T(v) ← Check load of the total route(v)
  if ($q_T(v) > Max_C$) **then**
    repair ← Last customers exceeding Max_C
$s' \leftarrow$ Move ***repair*** next to the customer with the minimum distance of the feasible routes to add the load
Return: s'

4. **Adjust Station Block:** If s' is feasible, or if the load is successfully repaired, the Adjust Station block is applied. This block

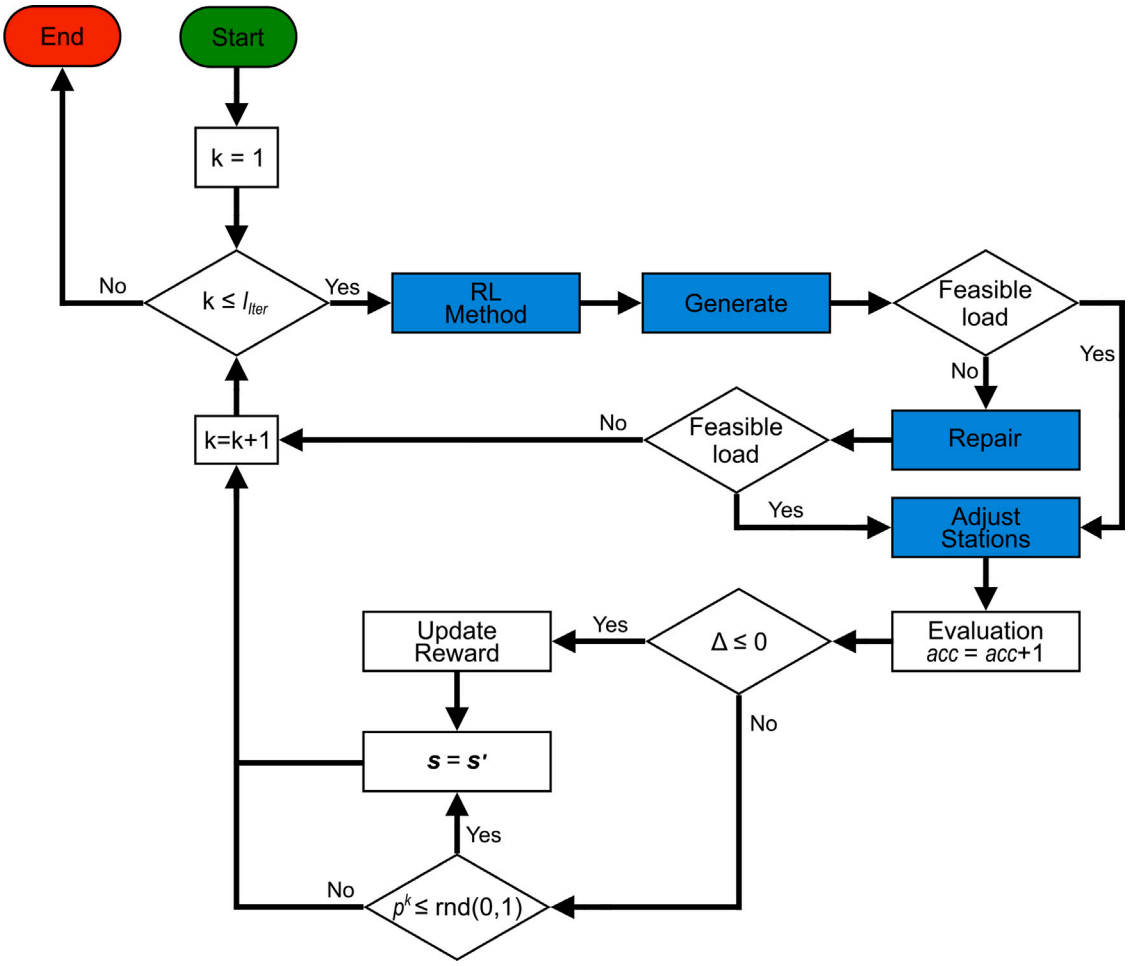


Fig. 3. Local search flowchart of HHASA_{RL}.

elucidated in detail in Algorithm 7, offers the flexibility to relocate or eliminate in a route a charging station based on probabilities using the roulette wheel selection method. The probabilities for relocation and elimination actions are denoted as p_r and p_e, respectively. It is essential to emphasize that stations are added solely to comply with power restrictions and guarantee that EVs do not run out of battery during the journey. Afterwards, the objective function is evaluated to obtain the total distance traveled by the vehicles. Δ is then calculated to decide whether to reward the selected heuristic for iteration k by updating the best solution or to use the Metropolis relation to accept movements with worse fitness. This local search process continues to iterate until iteration k reaches the maximum specified by I_{Iter}.

Algorithm 7 Pseudo-code of *AdjustStation* block

Inputs: s′
for (v ← 1 to num_{routes}) **do**
  e_T(v) ← Check the total energy of the route(v)
  if ($e_T(v) > Max_Q$) **then**
    s′ ← Add station where needed
  else
    if ($rand > 0.9$) **then**
      ii ← Roulette Wheel Selection (Relocate, Eliminate)
      s′ ← Apply(ii)
Return: s′

6. Experimental framework

In this section, we provide detailed information on the benchmark used, algorithms selected for comparison, and the statistical and non-parametric tests employed to assess the performance and efficiency of the proposed algorithm.

6.1. Benchmark description

The recent publicly available benchmark proposed for the IEEE WCCI2020 competition on computational intelligence for CEVRP is used to test the performance of the proposed approach (Mavrovouniotis et al., 2020b). This benchmark comprises 17 instances, including seven small instances with up to 100 customers and ten large instances with up to 1000 customers. The detailed information of each of the cases is summarized in Table 1.

The columns of the table present the number of customers (n_c), the number of charging stations (n_s) distributed in the space, the minimum number of EVs (Min_{Routes}), the maximum load of an EV (Max_C), the battery charge of an EV (Max_Q) and the energy consumption constant (h). The main objective is to minimize the total distance traveled by vehicles, with a single depot from which all EVs depart and return at the end of the route. It is worth noting that a solution may consist of multiple EVs to achieve this objective.

6.1.1. Baseline algorithms and compared methods

For comparisons, the analysis encompasses the three winning algorithms of the CEVRP competition at the IEEE WCCI2020 conference:

Table 1
Details of the CEVRP benchmark set.

Name	Customers (n_c)	Stations (n_s)	Min_{Routes}	Max_C	Max_Q	h
E22	21	8	4	6000	94	1.2
E23	22	9	3	4500	190	1.2
E30	29	6	4	4500	178	1.2
E33	32	6	4	8000	209	1.2
E51	50	5	5	160	105	1.2
E76	75	7	7	220	98	1.2
E101	75	9	8	200	103	1.2
X143	142	4	7	1190	2243	1.0
X214	213	9	11	944	987	1.0
X352	351	35	40	436	649	1.0
X459	458	20	26	1106	929	1.0
X573	572	6	30	210	1691	1.0
X685	684	25	75	408	911	1.0
X749	748	30	98	396	790	1.0
X819	818	25	171	358	926	1.0
X916	915	9	207	33	1591	1.0
X1001	1000	9	43	131	1684	1.0

VNS, SA, and GA. Additionally, the study incorporates the proposal by Woller et al. known as the Greedy Randomized Adaptive Search Procedure (GRASP) (Woller et al., 2020). Further comparison is drawn with the results presented by Jia et al. in 2021, who introduced the Bilevel Ant Colony Optimization (BACO) algorithm, demonstrating the best state-of-the-art results on this benchmark (Jia et al., 2021).

On the other hand, three versions of the proposed algorithm are compared using the simple methods for solving the multi-armed bandit problem, named HHASA$_{\epsilon-G}$, HHASA$_{UCB_1}$ and HHASA$_{TS}$. To assess the improvement of simple RL algorithms, a version that replaces the RL block with a random selection of heuristics, called HHASA, is employed.

6.2. Experimental setup

To ensure a fair comparison, the proposed approach adheres to all evaluation criteria specified in the competition. This involves conducting 20 independent runs with random seeds and applying a stopping criterion of $25,000 \cdot n_c$ maximum evaluations to the objective function (M_{Acc}), where n_c represents the instance size of the problem.

Within the set of internal parameters for HHASARL, α is assigned a value of 0.99, $limit$ is set to 20, I_{iter} is defined as $40 \cdot n_c$, and p_r and p_e are allocated 60% and 40%, respectively. Parameters used to obtain the β value in the reheating stage are 0% and 90% for x_{ini} and x_{end}, and 1 and 0.05 for y_{ini} and y_{end}, respectively. For heuristics, a value of 10% is selected for the subindex $r1$ and 100% for the subindex $r2$. All these parameters were empirically determined based on preliminary testing and observations of algorithm behavior.[1] Experiments were conducted using Matlab 9.4 on an Intel Core i5 CPU @ 2.7 GHz with 16 GB of RAM.

Since the source code of the proposed algorithms is not available for comparison to determine the efficiency of the proposed method, the results reported in the literature are used directly, which are the minimum (min), mean, and standard deviation (std) values over 20 runs. In addition, to assess that the statistical differences observed among the performance of the algorithms are statistically significant, Friedman's non-parametric test for multiple comparisons (Mousavirad et al., 2022; Scoczynski et al., 2021a) is employed, alongside Holm's post-hoc test for 1-to-n comparisons (Aziz et al., 2016).

7. Results and discussion

This section provides experimental results of the different aspects used to evaluate the efficiency of the HH proposal. First, a comparison is made with the statistical results of the independent runs of the proposals with the state-of-the-art. Then, the non-parametric analysis is shown to rank the algorithms and determine if there is a significant difference in their mean. Subsequently, a visual comparison of the selection of heuristics among the multi-armed bandit RL methods is presented. Finally, a comparison of the difference between the mean of the solutions against the best fitness found is reported.

7.1. Comparison of the statistical results

The statistical results obtained by applying all instances of the CEVRP benchmark to the algorithms mentioned in the previous section are shown in Tables 2 and 3. These tables present the min, mean, and std values of the solutions obtained by the algorithms in 20 independent runs. Bold and asterisk-marked values indicate the minimum average distances, while values presented in bold alone denote results surpassing the minimum mean obtained by the state-of-the-art.

According to the short instances in Table 2, the following observations can be obtained. According to the statistical test, the four variants of the proposed HH only show worse average results than BACO in instances E51 and E76. Other than that, HHASA$_{RL}$ and HHASA have equivalent performance with the compared algorithms and are superior for E33 and E101. Meanwhile, the HHASA$_{UCB1}$ algorithm achieves the lowest distance values for the objective function in the seven small instances. HHASA$_{\epsilon-G}$ obtains the best fitness in six cases with E101 missing, while HHASA$_{TS}$ and HHASA$_{TS}$ proposals found five. In general, results reveal that the HHASA$_{UCB1}$ algorithm is highly effective in finding the lowest fitness. However, it is challenging to determine which algorithm is better and more robust in these types of instances since HHASA$_{UCB1}$, HHASA$_{TS}$, and BACO present better mean results in five instances.

The results of comparing the algorithms on large instances are presented in Table 3. According to the findings, all four variants of the proposed HH demonstrate a lower mean value than the state-of-the-art results for the largest instances, which are X573, X685, X749, X819, X916, and X1001. Moreover, the proposed HHASA$_{TS}$ outperforms the instance X214, showcasing the best average fitness in six out of ten cases with a large number of customers and establishing itself as the most effective variant within the proposed HHs.

Regarding the minimum values found for the objective function, the HHASA$_{TS}$ algorithm updated the best-known solutions at five instances out of ten, namely X214, X685, X819, X916, and X1001. The HHASA$_{UCB1}$ algorithm also updated the fitness at X573 with a marginal difference of 0.90 compared to the best distance found by HHASA$_{TS}$, while the HHASA$_{\epsilon-G}$ updated the best fitness for the X459 instance and the HHASA updated the X749.

In summary, the results underscore the effectiveness of the HHASA$_{TS}$ approach in achieving the best mean values over 20 independent runs for large customer instances, demonstrating its capability to update minimum best-known solutions.

7.2. Non-parametric analysis

The results of the non-parametric analysis, conducted through the Friedman test for multiple comparisons and the Post Hoc Holm's test for 1-to-n comparisons, considering the 17 instances of the benchmark, are presented in Table 4. The p-values of Friedman's test, with values lower than 0.05, are highlighted in bold, suggesting the rejection of the null hypothesis of equal performance. Notably, the HHASA$_{TS}$ algorithm attains the best ranking according to the Friedman test.

It is worth mentioning that the three HH proposals that use multi-armed bandit RL methods have a better ranking than the HHASA with its random heuristic selection, indicating that the RL methods improve performance. Additionally, the p_{Holm} value of HHASA is less than 0.05, which implies that its performance is significantly worse than the control algorithm, HHASA$_{TS}$.

[1] The source code of the proposed algorithms is available at: https://github.com/erickre12/HHASARL.git.

Table 2
Results of the proposed algorithm applied to small instances of the benchmark.

Instances	Values	HHASA$_{TS}$	HHASA$_{UCB_1}$	HHASA$_{\epsilon-G}$	HHASA	BACO	VNS	SA	GA	GRASP
E22	min	384.67	384.67	384.67	384.67	384.67	384.67	384.67	384.67	389.82
	mean	**384.67***	**384.67***	**384.67***	**384.67***	**384.67***	**384.67***	**384.67***	**384.67***	389.89
	std	0	0	0	0	0	0	0	0	0.41
E23	min	571.94	571.94	571.94	571.94	571.94	571.94	571.94	571.94	571.94
	mean	**571.94***	**571.94***	**571.94***	572.51	**571.94***	**571.94***	**571.94***	**571.94***	572.36
	std	0	0	0	2.54	0	0	0	0	0.56
E30	min	509.47	509.47	509.47	509.47	509.47	509.47	509.47	509.47	512.19
	mean	**509.47***	**509.47***	**509.47***	**509.47***	**509.47***	**509.47***	**509.47***	**509.47***	512.67
	std	0	0	0	0	0	0	0	0	0.31
E33	min	840.14	840.14	840.14	840.14	840.57	840.14	840.57	844.25	841.08
	mean	840.70	**840.41***	840.82	841.10	842.30	840.43	854.07	845.62	845.06
	std	1.40	0.57	1.28	2.95	1.42	1.18	12.80	0.92	1.56
E51	min	529.90	529.90	529.90	529.90	529.90	529.90	533.66	529.90	536.09
	mean	536.98	535.13	534.16	535.09	**529.90***	543.26	533.66	542.08	546.21
	std	7.27	7.25	7.95	7.30	0	3.52	0	8.57	5.32
E76	min	692.74	692.64	692.64	694.54	692.64	692.64	701.03	697.27	701.63
	mean	694.96	695.65	695.14	694.94	**692.85***	697.89	712.17	717.30	711.36
	std	1.63	2.47	2.84	1.02	0.81	3.09	5.78	9.58	5.27
E101	min	837.10	835.63	836.17	837.10	840.25	839.29	845.84	852.69	847.47
	mean	**843.10***	**843.31**	**844.77**	**843.17**	845.95	853.34	852.48	872.69	856.86
	std	3.90	4.04	5.61	3.79	4.58	4.73	3.44	9.58	6.90

Table 3
Results of the proposed algorithm applied to large instances of the benchmark.

Instances	Values	HHASA$_{TS}$	HHASA$_{UCB_1}$	HHASA$_{\epsilon-G}$	HHASA	BACO	VNS	SA	GA	GRASP
X143	min	15 910.86	15 912.77	15 899.86	15 921.68	15 901.23	16 028.05	16 610.37	16 488.60	16 460.80
	mean	16 214.37	16 231.33	16 173.06	16 271.78	**16 031.46***	16 459.31	17 188.90	16 911.50	16 823.00
	std	215.77	173.73	198.91	250.09	262.47	242.59	170.44	282.30	157.00
X214	min	11 090.28	11 097.63	11 098.34	11 120.28	11 133.14	11 323.56	11 404.44	11 762.07	11 575.60
	mean	**11 206.60***	11 260.83	11 247.30	11 251.80	11 219.70	11 482.20	11 680.35	12 007.06	11 740.70
	std	84.58	88.73	99.53	73.03	46.25	76.14	116.47	156.69	80.41
X352	min	26 622.42	26 549.88	26 486.05	26 606.06	26 478.34	27 064.88	27 222.96	28 008.09	27 521.20
	mean	26 750.60	26 760.35	26 760.58	26 812.89	**26 593.18***	27 217.77	27 498.03	28 336.07	27 775.30
	std	102.55	116.44	135.77	90.44	72.86	86.20	155.62	205.29	111.99
X459	min	24 794.35	24 769.67	24 752.03	24 815.37	24 763.93	25 370.80	27 222.96	26 048.21	25 929.20
	mean	25 041.10	25 036.67	24 979.89	25 060.02	**24 916.60***	25 582.27	25 809.47	26 345.12	26 263.30
	std	237.58	114.68	151.54	121.58	94.08	106.89	157.97	185.14	134.66
X573	min	51 436.90	51 436.00	51 485.68	51 545.10	53 822.87	52 181.51	51 929.24	54 189.62	52 584.50
	mean	**51 776.70**	**51 764.24**	**51 771.50**	**51 748.42***	54 567.15	52 548.09	52 793.66	55 327.62	52 990.90
	std	166.86	152.69	158.01	119.57	231.05	278.85	577.24	548.05	246.79
X685	min	69 955.95	70 348.53	70 323.62	70 413.81	70 834.88	71 345.40	72 549.90	73 925.56	72 481.60
	mean	**70 401.25***	**70 719.10**	**70 684.34**	**70 791.10**	71 440.57	71 770.57	73 124.98	74 508.03	72 792.70
	std	218.98	291.53	174.26	222.85	281.78	197.08	320.07	409.43	189.53
X749	min	79 779.87	79 829.23	79 850.73	79 732.99	80 299.76	81 002.01	81 392.78	84 034.73	82 187.30
	mean	**80 135.67***	**80 256.36**	**80 318.42**	**80 397.82**	80 694.54	81 327.39	81 848.13	84 759.79	82 733.40
	std	219.50	303.30	399.47	432.11	223.91	176.19	275.26	376.10	213.21
X819	min	161 924.79	162 350.42	162 387.34	162 523.88	164 720.80	164 289.95	165 069.77	170 965.68	166 500.00
	mean	**162 530.67***	**162 819.78**	**162 883.17**	**163 031.19**	165 565.79	164 926.41	165 895.78	172 410.12	166 970.00
	std	289.41	258.35	300.11	389.12	401.02	318.62	403.70	568.58	211.84
X916	min	336 717.71	337 200.96	337 520.94	338 007.56	342 993.01	341 649.91	342 796.88	357 391.57	345 777.00
	mean	**337 641.92***	**338 349.57**	**338 639.53**	**338 688.50**	344 999.95	342 460.70	343 533.85	360 269.94	347 269.00
	std	461.47	454.28	544.69	328.64	905.72	510.66	556.98	229.19	654.93
X1001	min	75 469.29	75 864.07	75 782.95	75 850.15	76 297.09	77 476.36	78 053.86	78 832.90	77 636.20
	mean	**75 931.28***	**76 131.56**	**76 245.73**	**76 234.51**	77 434.33	77 920.52	NA	79 163.34	78 111.20
	std	304.10	212.24	226.30	271.18	719.86	234.73	306.27	NA	315.31

The following non-parametric analysis compares three HH proposals that contain the RL block and five state-of-the-art algorithms in Table 5. This table provides two comparisons; the first three columns consider all instances, while the remaining columns focus on 11 large instances from E101 to X1001.

In this analysis, it is evident that, for both comparisons, the top three ranking positions are secured by the HH proposals introduced in this work, with HHASATS emerging as the highest-ranked. Additionally, employing HHASATS as a control in the p_{Holm} analysis reveals a significant difference in performance compared to the VNS, SA, GRASP, and GA algorithms. Notably, while there is no statistically significant distinction with the BACO algorithm, the HHASA$_{TS}$ algorithm exhibits superior performance in the average results and overall ranking.

7.3. Analysis of the selection of heuristics of the HH proposals

Figs. 4 and 5 present plots detailing each local search within a single run for instances E101 and X916, respectively. The x-axis in both figures reflects the progression of iterations or local searches, while the y-axis in the first row signifies the frequency with which

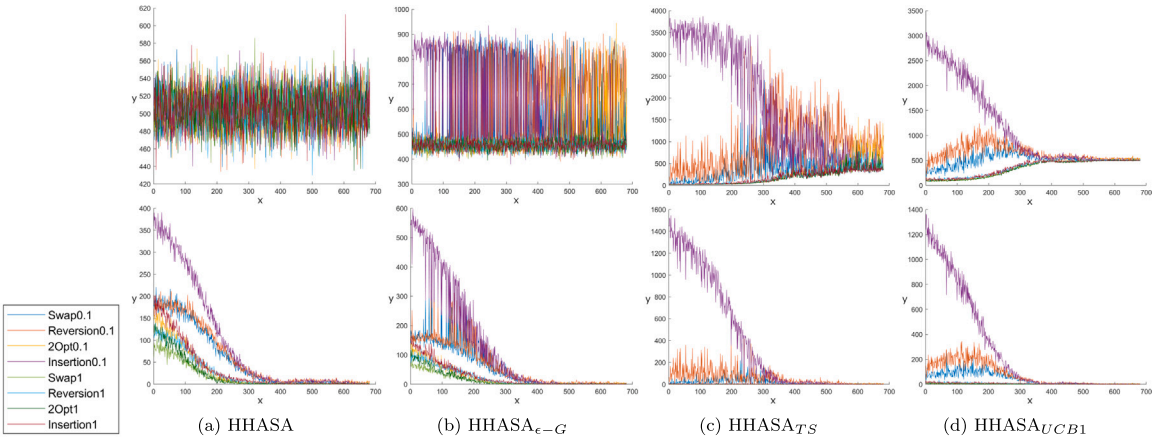


Fig. 4. Graphs of the vectors of the selected heuristics and the rewards of all the local searches of the four proposals of this work for the E101 instance.

Table 4
Average Friedman's rankings and Holm's p values (0.05) of the four proposed HHs for the CEVRP benchmark.

Algorithm	Ranking	p_{Holm}
HHASA$_{TS}$	1.8824	
HHASA$_{UCB1}$	2.4706	0.287886
HHASA$_{\epsilon-G}$	2.5294	0.287886
HHASA	3.1176	**0.015828**

Table 5
Average Friedman's rankings and Holm's p values (0.05) of the comparison with the state-of-the-art for the CEVRP benchmark.

Algorithm	Ranking $_{all}$	$p_{Holm\ all}$	Ranking $_{\geq E101}$	$p_{Holm\ \geq E101}$
HHASA$_{TS}$	2.4118		1.7273	
HHASA$_{UCB1}$	2.8824	0.882417	2.5455	0.676703
HHASA$_{\epsilon-G}$	3.0588	0.882417	2.7273	0.676703
BACO	3.4118	0.701859	3.5455	0.245168
VNS	4.6471	**0.031207**	4.8182	**0.012333**
SA	5.6471	**0.000589**	6.0909	**0.000147**
GRASP	6.8824	**0.000001**	6.6364	**0.000016**
GA	7.0588	**0**	7.9091	**0**

each heuristic was selected during the local search process. The second row on the y-axis presents the reward vector for the eight heuristics across all local searches. This reward vector illustrates how often each heuristic contributed to the improvement of the global solution when s' exhibited a fitness lower than s. Essentially, this vector serves as a quantitative measure of the effectiveness of each heuristic in enhancing the overall solution quality. The peaks and variations in the reward vector demonstrate the dynamics of heuristic selections during the local search process.

Figs. 4(a) and 5(a) show the HHASA proposal, and as expected, the heuristic selection vector does not give preference to any of them because it is random. Consequently, the reward figures showcase the performance of each heuristic when they have an equal chance of being chosen.

Similarly, Figs. 4(b) and 5(b) present the vectors of the HHASA$_{\epsilon-G}$ algorithm. Despite the similarity to the proposal without the RL method, notable differences exist, especially in the vector of selected heuristics. Peaks in this vector are more pronounced due to the nature of the ϵ-G algorithm, which chooses the heuristic with the highest reward in that local search.

The HHASA$_{TS}$ plots are depicted in Figs. 4(c) and 5(c). The Beta distribution used in the Thompson Sampling process evolves from a flat linear shape to a more realistic probability model of the mean reward as more data is collected. Actions with fewer trials have a higher range of possible values, contributing to wider dispersion. Consequently, a heuristic with a low estimated mean reward tried less frequently may yield a higher sample value, indicating its potential selection at that instant of the local search. In both cases, the $Insertion1$ heuristic is selected more frequently in the first half of the search process. However, in the second half of the search process, the heuristic selection graph begins to behave more like the random mode because the number of times a reward is obtained decreases.

Finally, in Figs. 4(d) and 5(d), the vectors of the proposed HHASA$_{UCB1}$ are presented graphically. The $UCB1$ method exhibits a lower regret level than the Epsilon Greedy and Thompson Sampling methods, enabling the quick identification of the optimal selection and testing of other heuristics only when uncertainty is high. The graphs show that in the beginning, $Insertion1$ is selected as the optimal heuristic at the end of the first local searches with fewer selections on the other heuristics. However, as the local searches progress, the uncertainty of $Insertion1$ increases, and more probability is assigned to the other heuristics to be selected.

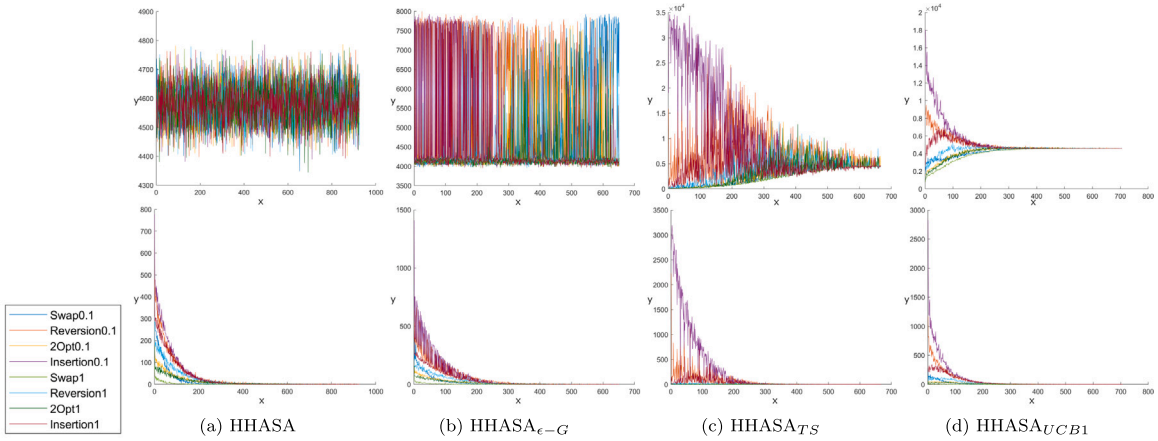
7.4. Comparison of additional energy used

The Table 6 illustrates the additional energy consumed by the solutions of the two most efficient HH proposals and the BACO algorithm. This extra energy is derived from the variance between the mean values obtained by each algorithm and the smallest distance found for the benchmark instances, as detailed in Tables 2 and 3. The energy consumption is calculated by multiplying the difference between the average and the best fitness, representing the average and shortest route lengths, respectively, by the energy consumption constant h (specified in Table 1). While the difference is not particularly noticeable for small instances, it becomes more significant as the number of clients increases, as observed in the X573 instance. Notably, the HHASA$_{TS}$ algorithm exhibits the lowest total energy difference in this comparison.

In Appendix B, a graphical analysis of HHASA$_{TS}$ solutions for selected CEVRP cases (E51, E101, X685, X1001) is presented. These visual representations provide insights into the performance of the algorithm across diverse complexities, showcasing strategic charging station placements and the effectiveness of the HH in addressing challenges posed by instances with a high number of customers.

8. Conclusions and future work

Last-mile logistics has had a significant economic, social, and environmental impact in urban areas, owing to the escalating number of vehicles engaged in goods transportation. This study aimed to formulate a methodology for optimizing freight vehicle routes by promoting the use of EVs to increase efficiency and reduce the times and/or costs of last-mile logistics. Consequently, an efficient algorithm named



(a) HHASA (b) HHASA$_{\epsilon-G}$ (c) HHASA$_{TS}$ (d) HHASA$_{UCB1}$

Fig. 5. Graphs of the vectors of the selected heuristics and the rewards of all the local searches of the four proposals of this work for the X916 instance.

Table 6
Difference in energy consumption between the average distance values and the minimum distance found so far.

Instances	HHASA$_{TS}$	HHASA$_{UCB1}$	BACO
E22	0	0	0
E23	0	0	0
E30	0	0	0
E33	0.67	0.32	2.59
E51	8.50	6.28	0.00
E76	2.78	3.61	0.25
E101	8.96	9.22	12.38
X143	313.14	330.10	130.23
X214	116.32	170.55	129.42
X351	272.26	282.01	114.84
X459	289.07	284.64	164.57
X573	340.70	328.24	3131.15
X685	445.30	763.15	1484.62
X749	402.68	523.37	961.55
X819	605.88	894.99	3641.00
X916	924.21	1631.86	8282.24
X1001	461.99	662.27	1965.04
Total	4192.47	5890.61	20 019.89

HHASA$_{RL}$ is introduced to address the optimization challenges of high-dimensional Capacitated Electric Vehicle Routing Problems (CEVRP), overcoming the limitations of existing state-of-the-art methods.

The proposed approach hybridizes the well-established Metropolis criterion from the self-adaptive SA metaheuristic as a movement acceptance mechanism and the RL algorithm as a heuristic selection mechanism. Experimental results showcase the superior performance of HHASARL on the IEEE WCCI2020 competition benchmark, outperforming all algorithms in the state-of-the-art employing the same dataset. Notably, the algorithm discovers multiple new best-known solutions for high-dimensional instances. The three proposals of HHASA$_{RL}$ algorithms have a more efficient and better performance than the compared algorithms for large instances. Among these, HHASA$_{TS}$ stands out as the top-performing algorithm, as indicated by average results and non-parametric tests, utilizing the Thompson Sampling method to address the multi-armed bandit problem.

It is essential to highlight that in this work, the distance was used as the objective function to test the efficiency of the proposed algorithm on large instances and to be able to compare on equal terms with the algorithms in the literature that have used the same benchmark. However, the objective function can be changed to decrease the routing time or tailored to suit specific applicability requirements.

According to the experiments conducted, although HHASA$_{TS}$ has shown better performance than the state-of-the-art algorithms, there are still multiple lines that deserve more research. In future work, we intend to modify the internal $AdjustStation$ block with a novel approach to improve the efficiency of predicting stops at the charging stations. In addition, the RL block will be replaced by deep RL methods to make it more robust and test the adaptability and efficiency of using these techniques as a heuristic selection mechanism.

Finally, it is planned to test this and the new approaches with the above improvements on more complex problems, such as the capacitated electric vehicle routing problem with time windows (CEVRPTW), where a fleet of delivery EVs must serve customers with known demand but with opening hours for a single product.

CRediT authorship contribution statement

Erick Rodríguez-Esparza: Conceptualization, Methodology, Software, Writing – original draft. **Antonio D. Masegosa:** Formal analysis, Supervision, Writing – review & editing. **Diego Oliva:** Writing – review & editing. **Enrique Onieva:** Visualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Antonio D Masegosa reports financial support was provided by University of Deusto.

Data availability

I have shared the link to my code in the manuscript.

Acknowledgments

This research has been partially funded by the University of Deusto Research Training Grants Programme, by the Spanish Ministry of Science and Innovation through the research project PID2022-140612OB-I00 and by the Basque Government through the research grants IT1564-22, KK-2023/00012 and KK-2023/00038. This research has also been partially supported by European Union's Horizon 2020 research and innovation programme under grant agreement No. 861540 [project SENATOR (Smart Network Operator Platform enabling Shared, Integrated and more Sustainable Urban Freight Logistics)].

Appendix A. Dynamic temperature control

Fig. A.6 provides a visualization of the behavior of the β dynamic temperature in local search optimization.

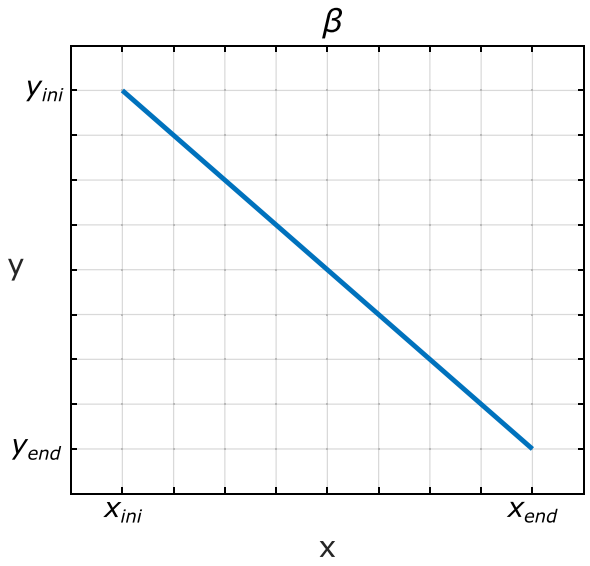


Fig. A.6. Temperature dynamics in the local search optimization.

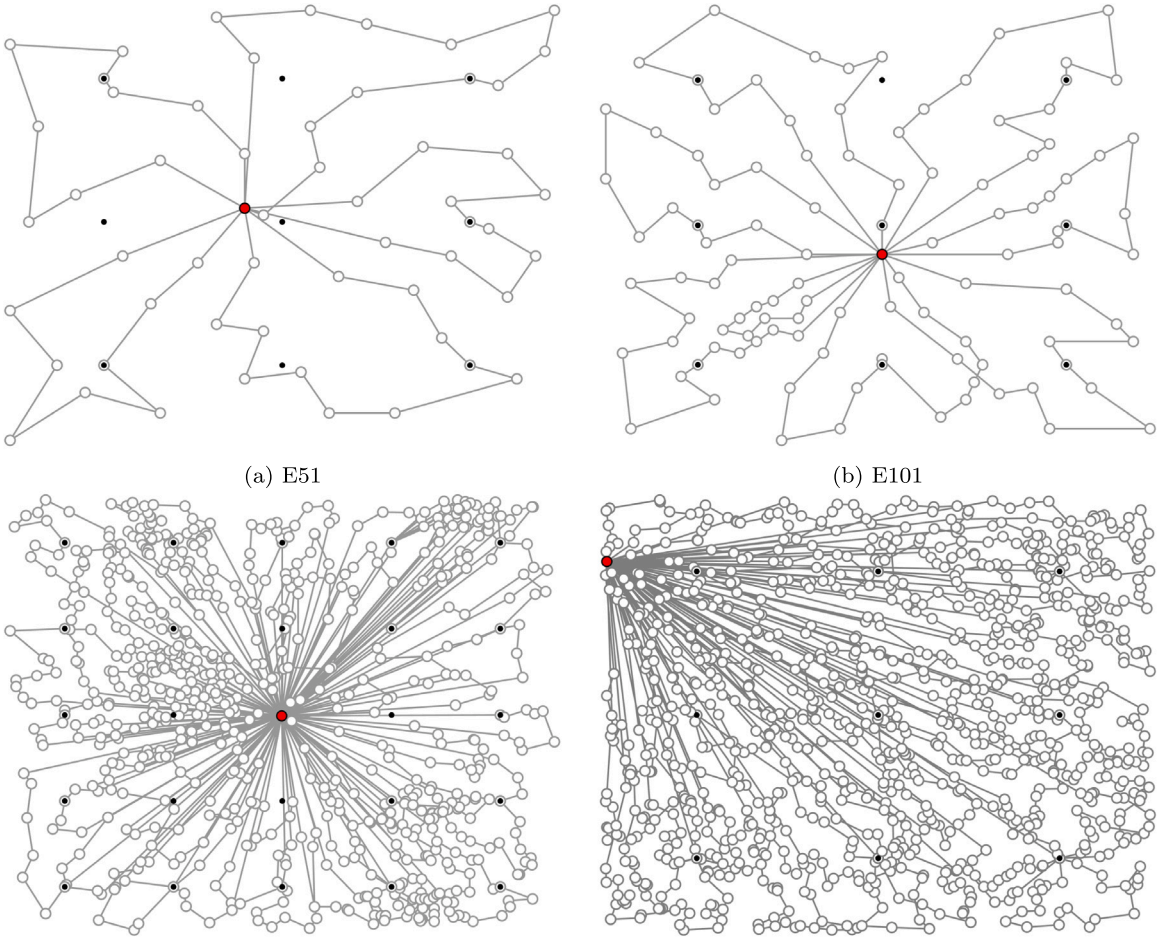
Appendix B. Graphical analysis of the solution

This appendix presents a graphical representation (Fig. B.7) intended to provide a visual understanding of the solutions generated by the HHASA$_{TS}$ algorithm for selected instances of the CEVRP benchmark. The chosen instances for visualization, namely E51, E101, X685, and X1001, are strategically selected to offer a comprehensive representation of different complexity levels within the benchmark. Notably, these instances exhibit varying n_c sizes, ranging from 50 to 1000, providing a nuanced perspective on the performance across a spectrum of problem intricacies.

For instances E51 and E101, as depicted in Figs. B.7(a) and B.7(b) respectively, the number of routes is visually appreciable. Fig. B.7(b) illustrates the eight routes identified in the best solution, highlighting examples where a charging station is left unused, and one route utilizes two charging stations.

In contrast, in Figs. B.7(c) and B.7(d), representing instances X685 and X1001 with larger problem sizes, it may be challenging to visualize the intricacies of individual routes due to the extensive routing distances. However, these instances are included to underscore the inherent complexity in benchmark problems of varying scales. The figures offer a visual representation of the challenging nature of the instances, enabling a qualitative understanding of the complexity without the necessity for detailed route visualization.

Furthermore, it is important to note that the proposed HH strategically places charging stations along the route, anticipating the depletion of EV batteries and ensuring timely and efficient replenishment to prevent the EVs from excessively deviating from the route to the remaining customers.



(a) E51 (b) E101

(c) X685 (d) X1001

Fig. B.7. Solutions generated by HHASA$_{TS}$ for instances E51, E101, X685, and X1001. The symbols •, ◦, and • represent the depot, customers, and charging stations, respectively.

References

Archetti, C., & Bertazzi, L. (2021). Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. *Networks*, *77*(2), 255–268.

Asghari, M., & e hashem, S. M. J. M. A. (2020). Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, Article 107899.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*(2), 235–256.

Aziz, N. A., Ibrahim, Z., Razali, S., & Aziz, N. A. A. (2016). Estimation-based meta-heuristics: A new branch of computational intelligence. In *The national conference for postgraduate research* (pp. 469–476).

Bhatti, A., Akram, H., Basit, H. M., Khan, A. U., Raza, S. M., & Naqvi, M. B. (2020). E-commerce trends during COVID-19 pandemic. *International Journal of Future Generation Communication and Networking*, *13*(2), 1449–1452.

Blocho, M. (2020). Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems. In *Smart delivery systems* (pp. 101–156). Elsevier.

Bogyrbayeva, A., Jang, S., Shah, A., Jang, Y. J., & Kwon, C. (2021). A reinforcement learning approach for rebalancing electric vehicle sharing systems. *IEEE Transactions on Intelligent Transportation Systems*.

Bosona, T. (2020). Urban freight last mile logistics—challenges and opportunities to improve sustainability: A literature review. *Sustainability*, *12*(21), 8769.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, *64*(12), 1695–1724.

Castillo, V. E., Bell, J. E., Rose, W. J., & Rodrigues, A. M. (2018). Crowdsourcing last mile delivery: Strategic implications and future research directions. *Journal of Business Logistics*, *39*(1), 7–25.

Choong, S. S., Wong, L.-P., & Lim, C. P. (2018). Automatic design of hyper-heuristic based on reinforcement learning. *Information Sciences*, *436*, 89–107.

de Santiago Júnior, V. A., Özcan, E., & de Carvalho, V. R. (2020). Hyper-heuristics based on reinforcement learning, balanced heuristic selection and group decision acceptance. *Applied Soft Computing*, *97*, Article 106760.

Delahaye, D., Chaimatanan, S., & Mongeau, M. (2019). Simulated annealing: From basics to applications. In *Handbook of metaheuristics* (pp. 1–35). Springer.

Drake, J. H., Kheiri, A., Özcan, E., & Burke, E. K. (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, *285*(2), 405–428.

Erdelić, T., & Carić, T. (2019). A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation*, *2019*.

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation research Part E: logistics and transportation review*, *48*(1), 100–114.

Fafoutellis, P., Mantouka, E. G., & Vlahogianni, E. I. (2021). Eco-driving and its impacts on fuel efficiency: An overview of technologies and data-driven methods. *Sustainability*, *13*(1).

Fausto, F., Reyna-Orta, A., Cuevas, E., Andrade, Á. G., & Perez-Cisneros, M. (2020). S. *Artificial Intelligence Review*, *53*(1), 753–810.

Gittins, J., Glazebrook, K., & Weber, R. (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons.

Giuffrida, N., Fajardo-Calderin, J., Masegosa, A. D., Werner, F., Steudter, M., & Pilla, F. (2022). Optimization and machine learning applied to last-mile logistics: A review. *Sustainability*, *14*(9), 5329.

He, J., Yang, H., Tang, T.-Q., & Huang, H.-J. (2018). An optimal charging station location model with the consideration of electric vehicle's driving range. *Transportation Research Part C (Emerging Technologies)*, *86*, 641–654.

Ignat, B., & Chankov, S. (2020). Do e-commerce customers change their preferred last-mile delivery based on its sustainability impact? *The International Journal of Logistics Management*.

Janjevic, M., Knoppen, D., & Winkenbach, M. (2019). Integrated decision-making framework for urban freight logistics policy-making. *Transportation Research Part D: Transport and Environment*, *72*, 333–357.

Jia, Y.-H., Mei, Y., & Zhang, M. (2021). A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. *IEEE Transactions on Cybernetics*.

Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C (Emerging Technologies)*, *65*, 111–127.

Keskin, M., & Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers & Operations Research*, *100*, 172–188.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Largo, S., Souissi, O., & El Akkaoui, Z. (2020). Green vehicle routing problem: A short survey. In *2020 IEEE international conference on technology management, operations and decisions* (pp. 1–10). IEEE.

Lin, B., Ghaddar, B., & Nathwani, J. (2020). Deep reinforcement learning for electric vehicle routing problem with time windows. arXiv preprint arXiv:2010.02068.

Mao, H., Shi, J., Zhou, Y., & Zhang, G. (2020). The electric vehicle routing problem with time windows and multiple recharging options. *IEEE Access*, *8*, 114864–114875.

Mavrovouniotis, M., Ellinas, G., & Polycarpou, M. (2018). Ant colony optimization for the electric vehicle routing problem. In *2018 IEEE symposium series on computational intelligence* (pp. 1234–1241). IEEE.

Mavrovouniotis, M., Menelaou, C., Timotheou, S., Ellinas, G., Panayiotou, C., & Polycarpou, M. (2020a). A benchmark test suite for the electric capacitated vehicle routing problem. In *2020 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.

Mavrovouniotis, M., Menelaou, C., Timotheou, S., Panayiotou, C., Ellinas, G., & Polycarpou, M. (2020b). *Benchmark set for the IEEE WCCI-2020 competition on evolutionary computation for the electric vehicle routing problem*: *Tech. rep.*, KIOS CoE, University of Cyprus, Cyprus.

Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2016). A multi-space sampling heuristic for the green vehicle routing problem. *Transportation Research Part C (Emerging Technologies)*, *70*, 113–128.

Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research, Part B (Methodological)*, *103*, 87–110.

Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist? *Swarm and Evolutionary Computation*, *54*, Article 100671.

Morales-Castaneda, B., Zaldivar, D., Cuevas, E., Maciel-Castillo, O., Aranguren, I., & Fausto, F. (2019). An improved simulated annealing algorithm based on ancient metallurgy techniques. *Applied Soft Computing*, *84*, Article 105761.

Mousavirad, S. J., Oliva, D., Chakrabortty, R. K., Zabihzadeh, D., & Hinojosa, S. (2022). Population-based self-adaptive generalised masi entropy for image segmentation: A novel representation. *Knowledge-Based Systems*, Article 108610.

Oliva, D., Rodriguez-Esparza, E., Martins, M. S., Abd Elaziz, M., Hinojosa, S., Ewees, A. A., & Lu, S. (2020). Balancing the influence of evolutionary operators for global optimization. In *2020 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.

Osaba, E., Yang, X.-S., & Del Ser, J. (2020). Is the vehicle routing problem dead? an overview through bioinspired perspective and a prospect of opportunities. In *Nature-inspired computation in navigation and routing problems* (pp. 57–84). Springer.

Patella, S. M., Grazieschi, G., Gatta, V., Marcucci, E., & Carrese, S. (2021). The adoption of green vehicles in last mile logistics: A systematic review. *Sustainability*, *13*(1), 6.

Pelletier, S., Jabali, O., & Laporte, G. (2016). 50Th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transportation science*, *50*(1), 3–22.

Pillay, N., & Qu, R. (2018). *Hyper-heuristics: Theory and applications*. Springer.

Purkayastha, R., Chakraborty, T., Saha, A., & Mukhopadhyay, D. (2020). Study and analysis of various heuristic algorithms for solving Travelling Salesman Problem—A survey. In *Proceedings of the global AI congress 2019* (pp. 61–70). Springer.

Rodríguez-Esparza, E., Morales-Castañeda, B., Casas-Ordaz, A., Oliva, D., Navarro, M. A., Valdivia, A., & Houssein, E. H. (2024). Handling the balance of operators in evolutionary algorithms through a weighted hill climbing approach. *Knowledge-Based Systems*, Article 111784.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2017). A tutorial on Thompson sampling. arXiv preprint arXiv:1707.02038.

Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, *48*(4), 500–520.

Scoczynski, M., Delgado, M., Lüders, R., Oliva, D., Wagner, M., Sung, I., & El Yafrani, M. (2021). Saving computational budget in Bayesian network-based evolutionary algorithms. *Natural Computing*, *20*(4), 775–790.

Scoczynski, M., Oliva, D., Rodríguez-Esparza, E., Delgado, M., Lüders, R., Yafrani, M. E., Ledo, L., Elaziz, M. A., & Peréz-Cisnero, M. (2021). A selection hyperheuristic guided by Thompson sampling for numerical optimization. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1394–1402).

Shi, J., Gao, Y., Wang, W., Yu, N., & Ioannou, P. A. (2019). Operating electric vehicle fleet for ride-hailing services with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, *21*(11), 4822–4834.

Singh, S., Kumar, R., Panchal, R., & Tiwari, M. K. (2021). Impact of COVID-19 on logistics systems and disruptions in food supply chain. *International Journal of Production Research*, *59*(7), 1993–2008.

Slivkins, A. (2019). Introduction to multi-armed bandits. arXiv preprint arXiv:1904.07272.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265.

Swiercz, A. (2017). Hyper-heuristics and metaheuristics for selected bio-inspired combinatorial optimization problems. In *Heuristics and hyper-heuristics-principles and applications*. IntechOpen.

Turky, A., Sabar, N. R., Dunstall, S., & Song, A. (2018). Hyper-heuristic based local search for combinatorial optimisation problems. In *Australasian joint conference on artificial intelligence* (pp. 312–317). Springer.

Umami, I., & Rahmawati, L. (2021). Comparing Epsilon Greedy and Thompson sampling model for multi-armed bandit algorithm on marketing dataset. *Journal of Applied Data Sciences*, *2*(2).

Vakulenko, Y., Shams, P., Hellström, D., & Hjort, K. (2019). Service innovation in e-commerce last mile delivery: Mapping the e-customer journey. *Journal of Business Research*, *101*, 461–468.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, *231*(1), 1–21.

Viu-Roig, M., & Alvarez-Palau, E. J. (2020). The impact of E-commerce-related last-mile logistics on cities: A systematic literature review. *Sustainability*, *12*(16), 6492.

Wang, Q., & Tang, C. (2021). Deep reinforcement learning for transportation network combinatorial optimization: A survey. *Knowledge-Based Systems*, *233*, Article 107526.

Woller, D., Kozák, V., & Kulich, M. (2020). The GRASP metaheuristic for the electric vehicle routing problem. In *International conference on modelling and simulation for autonomous systems* (pp. 189–205). Springer.

Yang, T., Zhang, S., & Li, C. (2021). A multi-objective hyper-heuristic algorithm based on adaptive epsilon-greedy selection. *Complex & Intelligent Systems*, *7*(2), 765–780.

Yi, Z., Smart, J., & Shirk, M. (2018). Energy impact evaluation for eco-routing and charging of autonomous electric vehicle fleet: Ambient temperature consideration. *Transportation Research Part C (Emerging Technologies)*, *89*, 344–363.

Zhang, Y., Bai, R., Qu, R., Tu, C., & Jin, J. (2021). A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research*.

Zhao, J., Mao, M., Zhao, X., & Zou, J. (2020). A hybrid of deep reinforcement learning and local search for the vehicle routing problems. *IEEE Transactions on Intelligent Transportation Systems*.

Zirour, M. (2008). Vehicle routing problem: Models and solutions. *Journal of Quality Measurement and Analysis JQMA*, *4*(1), 205–218.