# A Review About Electric Vehicle Routing Problem with Reinforcement Learning

**Abdelkader Kaddour*[1], Lamri Sayad[2]**
[1]Department of Computer Science, Faculty of Mathematics and Computer Science, University of M'sila, Algeria
[2]Laboratory of Informatics and its Applications of M'sila (LIAM), Faculty of Mathematics and Computer Science, University of M'sila, Algeria
**(Corresponding Author):*E-mail:** [1] abdelkader.kaddour@univ-msila.dz/kaddour.abdelkader.pro@gmail.com

**Received:** 15-02-2023 **Accepted:** 20-07-2023 **Published:** 09-08-2023

**Abstract**
The Electric Vehicle Routing Problem (EVRP) is a variant of the traditional Vehicle Routing Problem (VRP) that deals explicitly with the routing and scheduling of electric vehicles (EVs). It considers EVs' unique constraints and characteristics, such as limited driving range and the need for battery charging. Reinforcement Learning (RL) is a type of machine learning that involves training an agent to make a series of decisions in an environment to maximize a reward. RL has been successfully applied to various problems, including game-playing, robotics, and decision-making under uncertainty. Some key challenges in RL include dealing with large state and action spaces, balancing exploration and exploitation, and dealing with non-stationary environments. RL has emerged as a promising approach for solving the EVRP in recent years. In the context of the EVRP, the agent could be an electric vehicle, and the environment could be a city with charging stations and customer locations. The agent's decisions encompass selecting the most optimal routes and undertaking specific actions. The reward could measure the efficiency and cost-effectiveness of the routes taken. RL can find near-optimal solutions to the EVRP in a more flexible and adaptable way than traditional optimization methods. In this review article, we will discuss the application of RL to the EVRP, the challenges, and opportunities of using RL for this problem and its variants, the current state of the art in RL-based approaches for the EVRP, and directions for future research.
**Keywords:** Electric Vehicle Routing Problem (EVRP), Electric Vehicles (EVs), Reinforcement Learning (RL).

**Tob Regul Sci. ™ 2023;9(1): 4175-4192**
**DOI: doi.org/10.18001/TRS.9.293**

## 1. Introduction

Creating an ecologically friendly transportation system is critical in developing smarter cities. We are now implementing a new technological solution, such as electric mobility, for adequate urban transportation, which helps to reduce hazardous emissions. Several prominent organizations are already considering using commercial EVs in their day-to-day operations in the logistics industry.

However, given the characteristics of this new technology, they will need to develop practical route-planning tools. We have recently witnessed continuous growth in Europe's energy expenditures and hazardous gas emissions. These factors and society's growing economic, environmental, and social consciousness have sparked many business green initiatives. In today's economy, marketplaces are becoming more open and competitive. A freight transportation system that is efficient, sustainable, and environmentally friendly is a critical success factor. Modern transportation businesses are worried about fuel costs and environmental degradation caused by flows of freight's greenhouse gas (GHG) emissions.

Transportation is one of the primary beneficiaries of green logistics operations. As a result, it strives to improve the sustainability of production and distribution processes by considering environmental and social concerns [1]. The authors of [2] discovered that the transport sector is responsible for 30% of CO2 emissions in the EU, with urban regions accounting for 40% of CO2 emissions. Furthermore, according to the United States Environmental Protection Agency (2016), the transportation sector accounts for 30% of GHG emissions in the United States [3]. Building efficient, faster, and less fuel-consuming transportation networks have been a significant issue in recent years. Renewable energy technology and efficient transportation operations are two primary approaches to sustainable transportation. Fuel Vehicles (AFVs) were replaced by hybrid vehicles (HVS), plug-in hybrid electric cars (PHEVs), and electric vehicles (EVs) to develop renewable energy technology. Road freight transport, including city logistics, accounts for 33% of transport emissions [4].

The Electric Vehicle Routing Problem (EVRP) addresses the routing and scheduling of electric vehicles (EVs). It considers these vehicles' unique constraints and characteristics, such as limited driving range and the need for battery charging. The EVRP aims to find the most efficient and cost-effective routes for EVs to follow to serve a set of predetermined customer locations while ensuring that the vehicles do not run out of power. The EVRP has gained significant attention in recent years due to the increasing adoption of EVs and the need for efficient and sustainable transportation systems. The use of EVs can help reduce greenhouse gas emissions and air pollution, but their widespread adoption is hindered by the limited driving range and availability of charging infrastructure. Developing efficient routing strategies for EVs is an important research topic.

In recent years, reinforcement learning (RL) has emerged as a promising approach for solving the EVRP. RL is a type of machine learning that involves training an agent to make a series of decisions in an environment to maximize a reward. In the context of the EVRP, the agent could be an EV, and the environment could be a city with charging stations and customer locations. The agent's decisions would be the routes it takes and its actions (e.g., charging the battery at a certain station). The reward could measure the efficiency and cost-effectiveness of the routes taken. RL can find near-optimal solutions to the EVRP in a more flexible and adaptable way than traditional optimization methods. There are several challenges and opportunities in applying RL to the EVRP. One challenge is the ample state space of the problem, which can

make it difficult for the RL agent to learn and explore effectively. Another challenge is the dynamic nature of the EVRP, as the availability and demand for charging stations can vary over time. The RL agent must adapt and learn from experience to make effective routing decisions. On the other hand, using RL can enable more intelligent and autonomous EV routing, leading to more efficient and sustainable transportation systems. The remainder of this paper is structured as follows: In section (2), we will introduce the details of the EVRP. Section (3) will present the RL and its types. Section (4) discusses previous work that used RL for EVRP. Finally, we will conclude the work with insight for future work in section (5).

## 2. Electric Vehicle Routing Problem

The EVRP is a variant of the Vehicle Routing Problem (VRP) in which the vehicles are electric and have limited range. The EVRP is the process of finding a set of vehicle routes. Each route services a set of customer nodes and starts and ends at a given depot node. The problem aims to find the best route plan for electric vehicles that minimizes a given cost function while satisfying several restrictions and operational procedures for electric vehicles. According to existing studies, the basic assumptions for the EVRP are as follows [5, 6, 7]:

- Each route starts and ends at the depot node.
- Each customer node is to be serviced by exactly one electric vehicle.
- Electric vehicles can visit a charging station to recharge operations between customers.
- Each charging station can be visited by more than one electric vehicle.
- The locations of the charging stations and the traveling distances from any node (i.e., origin or destination) to any charging station are known.
- The battery level of an electric vehicle must always be between 0 and its battery capacity.
- A vehicle's battery is always fully charged when visiting a charging station.

Following the assumptions above, Figure 1 presents an illustrative example of a solution to the EVRP involving 15 customer nodes (C1, ..., C15), five charging stations (S1, ..., S5), and the depot node that can also be used as a charging station. Four identical electric vehicles serve customer nodes by starting their tour at the depot node with a full charge. The percentage values on the arcs show the battery level of the electric vehicle when it arrives at a customer location or the depot node. Additionally, since the vehicles are fully charged at stations, battery levels after charging station visits are set to 100%. In addition to the basic EVRP assumptions, other commonly used restrictions come from vehicle capacity constraints and time-related restrictions. Vehicle weight or volume capacity can be considered a constraint where the total weight or volume of the loads cannot exceed the vehicle's weight or volume capacity, respectively. Several assumptions exist for the time-related restrictions that can be summarized into two groups: time windows for nodes and duration time limits. Time window restrictions state that each customer node must be serviced within a given time window, and each route must be completed within a given time window limit of a depot node [8, 9, 10]. Time duration constraints state that the total elapsed time for a route cannot exceed the duration time limit [6, 11]. Similar to the VRP with

time windows (VRPTW), arc travel times, customer service times, time windows are given beforehand, and vehicle travel and waiting. Service times can be determined similarly as in the VRPTW. In addition, the recharging time at charging stations is computed using a function or constant value [3, 12, 13]. When time-related constraints are considered for the EVRP, station charging operations become more critical. Therefore, partial charging of electric vehicles is also studied in most of the papers[14].

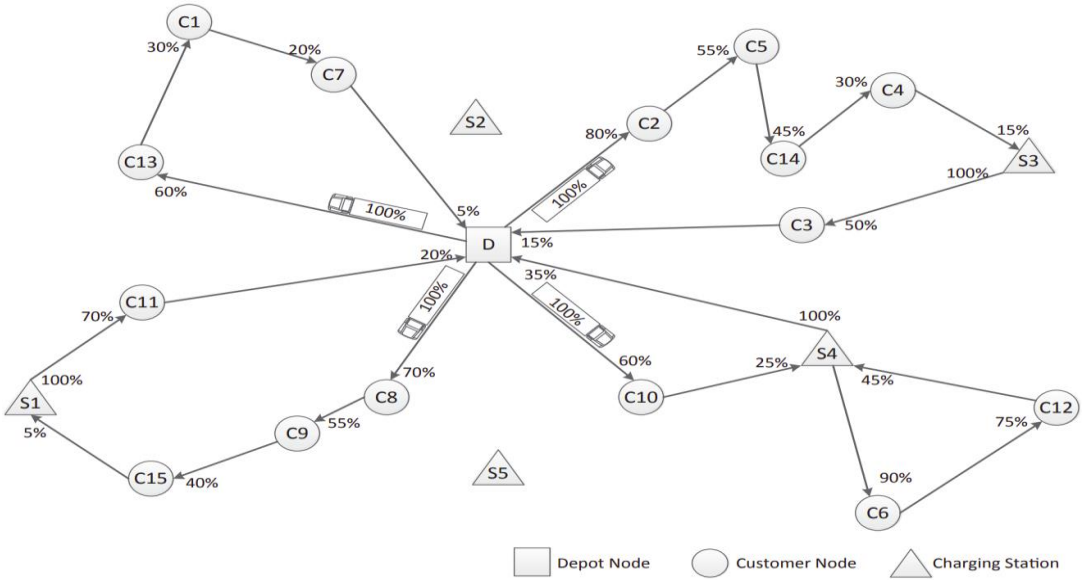


**Figure 1: An illustrative example of the EVRP [14].**

The EVRP aims to determine the optimal routes for a fleet of electric vehicles to serve a set of customer demand locations while considering the vehicles' battery capacity constraints and the availability of charging stations. Several variants of the EVRP have been proposed in the literature to address different specific aspects of the problem:

**Capacitated EVRP (CEVRP):** where each vehicle has a fixed capacity to carry goods or passengers [15].

- **Time-Dependent EVRP (TD-EVRP):** where the customer demand, vehicle speed, and availability of charging stations vary over time [16].
- **The EVRP with Time Windows (EVRPTW):** where customers have specific time windows for service, and vehicles must arrive within those windows [10].
- **Multi-Depot EVRP (MD-EVRP):** where the vehicles can start and end at different depots[17].
- **Dynamic Stochastic EVRP (DS-EVRP):** where the vehicles operate in a dynamic and stochastic environment [18].
- **Dynamic EVRP (D-EVRP):** where the customer demand, travel times, and charging station availability change over time [19].
- **EVRP with Battery Swapping (EVRP-BS):** where electric vehicles have the option to replace their depleted batteries with fully charged ones at designated stations [20].
- **EVRP with Pickup and Delivery (EVRPPD):** where the vehicles pick up and deliver goods or passengers at different locations [21].

- **EVRP with flexible deliveries:** where the customers are served using a fleet of EVs that can recharge their batteries along their routes. A customer may specify different delivery locations for different time windows in this problem [22].
- **EVRP with Parking Constraints (PC-EVRP):** where routing decisions consider parking constraints at the charging stations and customer locations [23].
- **Two-echelon EVRP (2E-EVRP):** In the first echelon, fossil fuel-powered trucks transport goods from the depot to a subset of satellites. The second echelon transfers goods from the satellites to the final customers using EVs [24].

## 3. Reinforcement Learning

RL is a machine learning type involving training an agent to take action in an environment to maximize a reward [25]. It is based on learning by trial and error, where the agent receives rewards or penalties for its actions and uses this feedback to adjust its behavior over time. RL algorithms use optimization techniques to learn the best actions in a given environment to maximize the reward, which involves estimating the value of each activity based on its expected future rewards and choosing the action with the highest value [26]. Standard optimization algorithms used in RL include dynamic programming, temporal difference learning, and Monte Carlo methods. In addition to learning the best actions, optimization is often used in designing RL algorithms. For example, the learning rate selection determines how much the agent updates its estimates of action values based on new information, which is a vital optimization problem in RL. The learning rate must be carefully chosen to balance the trade-off between exploration (trying out new actions to learn more about the environment) and exploitation (taking the best-known actions to maximize the reward). Overall, the combination of RL and optimization enables agents to learn and adapt to their environments in real-time, making them highly effective at solving complex problems in various applications.

### 3.1 Components of RL

RL consists of several key components, which include [25]:

- **Agent:** The agent is the entity that interacts with the environment and makes decisions based on the observations it receives. The agent aims to learn a policy that maximizes the expected cumulative reward over time.
- **Environment:** The environment is the world or system that the agent interacts with. The environment provides the agent with observations, and in response to the agent's actions, it transitions to a new state and provides the agent with a scalar reward signal.
- **State:** A state describes the environment at a particular point in time. The state can be fully observable, partially observable, or even unobservable. The agent's policy, in RL, is mapping states to actions.
- **Action:** An action is an agent's choice in response to its current state. The environment determines the set of actions available to the agent.

- **Reward:** The reward is a scalar signal the environment provides to the agent in response to its actions. The agent aims to learn a policy that maximizes the expected cumulative reward over time.
- **Policy:** Policy is the decision-making strategy or mapping of the agent that describes how it will act based on the state of the environment. The policy can be deterministic or probabilistic.
- **Value function:** The value function assigns a scalar value to each state or state-action pair, representing the expected cumulative reward of being in that state or taking that action. It estimates how good or bad a particular state or action is. There are two types of value functions, State-Value and Action-Value.
- **Model of the environment:** A model is an approximation of the environment that the agent can use to simulate the effects of its actions. This approach can be used to plan actions rather than relying on trial and error.

Figure 2 demonstrates the action-reward feedback loop of a generic RL model.

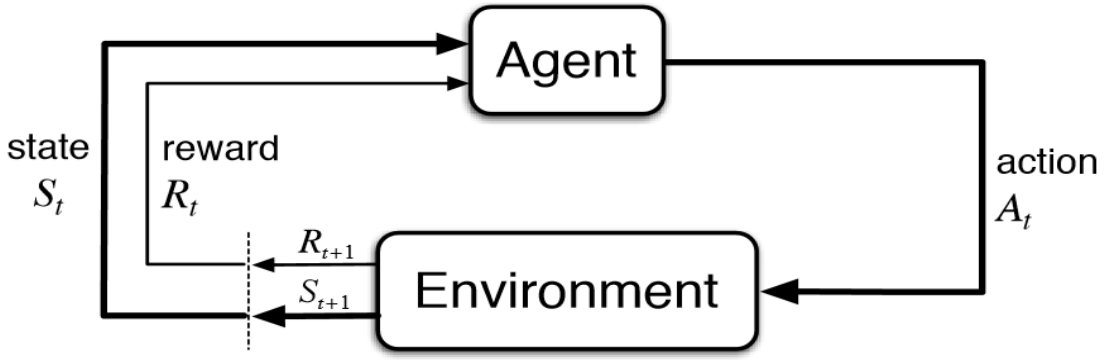


**Figure 2: An illustrative example of the generic RL model.**

3.2 types of RL

- **Model-based RL:** In model-based RL, the agent acquires a model of the environment, enabling it to plan actions. This approach offers greater sample efficiency than model-free RL since the agent can predict action outcomes and plan accordingly. Nonetheless, model-based RL is more vulnerable to errors in the acquired model, as such errors can propagate during planning, resulting in suboptimal or potentially harmful policies [27].
- **Model-free RL:** In model-free RL, the agent does not learn a model of the environment but instead learns a policy directly from experience. This technique can be less sample efficient than model-based RL, as the agent must interact with the environment to gather data and learn the policy. However, model-free RL is also more robust to errors in the model, as the agent is not reliant on a potentially inaccurate environment model [27].
- **Off-policy RL:** In off-policy RL, the agent learns a policy different from the one it currently follows. This approach proves beneficial in understanding the long-term consequences of actions, as the agent gains insights into the outcomes of actions it does not undertake [28]. Furthermore,

off-policy RL can simultaneously learn about multiple policies by acquiring a value function that assesses the relative return of different policies [29].

- **Deep RL:** It involves utilizing deep neural networks to learn policies or value functions within the context of RL. This approach has achieved remarkable success across diverse applications, including games, natural language processing, and robotics [30]. Deep RL algorithms demonstrate the capacity to learn intricate policies or value functions, which would be challenging to design manually, and they can effectively learn from raw sensory data, such as images or audio.
- **Multi-agent RL:** It entails multiple agents interacting with each other and the environment. This domain can be categorized into cooperative and competitive scenarios, depending on whether the agents collaborate towards a shared goal or compete [30]. Multi-agent RL presents additional challenges compared to single-agent RL, as the agents may have conflicting objectives or require coordination of actions to achieve mutual goals [31].
- **Hierarchical RL:** In hierarchical RL, the learning problem is decomposed into multiple levels of subproblems, with each level representing a different time scale or abstraction level. This approach can be advantageous in cases where the environment is too complex to be modeled directly or when multiple conflicting goals need to be balanced [32]. Hierarchical RL can also be more sample efficient than flat RL, as the agent can learn about lower-level subproblems independently and reuse this knowledge when learning higher-level tasks [33].
- **Transfer learning:** Transfer learning in RL involves utilizing knowledge acquired from one task to enhance learning in a distinct yet related task. This strategy proves valuable in cases where it is expensive or difficult to collect data for the target task or when a large amount of related data can be used to learn a good policy. Transfer learning in RL can be challenging, as the differences between the source and target tasks may be significant, and the agent may need to adapt its learned knowledge to the new task [34].
- **Imitation learning:** It refers to using demonstrations or expert data to learn a policy or value function. This approach can be practical when it is difficult or expensive to specify a reward function or when existing experts or demonstrators can provide data [35]. Imitation learning can be further divided into behavioral cloning, in which the agent learns to mimic the demonstrator's behavior, and inverse RL, in which the agent learns a reward function or cost function from the demonstrator's data [36].
- **Evolutionary RL:** Evolutionary RL entails the utilization of evolutionary algorithms, such as genetic algorithms, to learn policies or value functions. This approach can be advantageous in scenarios where the environment is stochastic or dynamic, as the evolutionary algorithm can explore robust policies that perform well across varying conditions. Furthermore, Evolutionary RL is adept at learning complex policies that would be challenging to design manually [37].
- **Online RL:** In online RL, the agent learns from a sequence of interactions with the environment without prior knowledge of the transition dynamics or reward function. Online RL can be proper in cases where the environment is changing or non-stationary, as the agent can adapt to

the changing conditions on the fly [25]. Online RL can also be more sample efficient than offline RL, as the agent can learn from each interaction and immediately use this knowledge to improve its policy.

- **Offline RL:** Offline RL refers to using offline or batch data to learn a policy or value function. It can be useful in cases where it is expensive to collect data online or when there is a large amount of data available that has already been collected [25].
- **Continuous control RL:** Continuous control RL refers to using RL techniques to control systems with continuous action spaces. It can be challenging, as the action space is often unbounded, and the optimization problem may be non-convex [38]. Continuous control RL algorithms often use function approximators, such as neural networks, to learn policies or value functions, and may use techniques such as action repeats or action noise to improve exploration [39].
- **Partially observable RL:** In partially observable RL, the agent must learn a policy based on a limited and noisy view of the environment. It can be challenging, as the agent may need to use its memory or other internal state to reason about the underlying state of the environment [25]. Partially observable RL algorithms often use Bayesian filtering or particle filtering techniques to estimate the underlying state of the environment. They may use recurrent neural networks or other forms of memory to store and manipulate this state [40].
- **Multi-objective RL:** Navigating conflicting objectives can pose a considerable challenge to the agent, as it involves intricate trade-offs between different goals and determining their relative importance. Multi-objective RL algorithms often adopt Pareto optimization or scalarization techniques to balance these diverse objectives. Additionally, interactive methods enable the user to specify the relative significance of different objectives [41].
- **Safe RL:** Safe RL involves utilizing RL techniques to acquire safe policies that satisfy constraints. This technique becomes crucial when the agent operates in real-world environments and must avoid harmful or undesirable actions [42].
- **Inverse RL:** Inverse RL, also known as apprenticeship learning, refers to the problem of learning a reward function or cost function from expert demonstrations [43]. It can be useful in cases where it is difficult or expensive to specify a reward function manually or when existing experts or demonstrators can provide data. Inverse RL algorithms often use maximum entropy inverse RL or inverse optimal control techniques to learn the reward function from the expert's data.
- **Hybrid methods:** Hybrid methods refer to the combination of RL with other machine learning techniques, such as supervised learning, unsupervised learning, or planning. Hybrid methods can be helpful in cases where RL alone is insufficient to solve the problem or when combining RL with other techniques leads to improved performance [44]. Examples of hybrid methods include Deep RL with imitation learning or RL with planning and searching.
- **Adversarial RL:** Adversarial RL involves employing RL techniques in scenarios where the agent and the environment are in conflict or competition. This classification encompasses two categories: (1) adversarial training, where the agent optimizes its reward by adapting to a fixed or

learned environment model, and (2) multiagent RL, where multiple agents compete or collaborate in the environment [36]. Adversarial RL presents challenges, as the agent must learn to anticipate and counter the adversary's actions.

- **RL with side information:** It utilizes extra data, like natural language descriptions or expert knowledge, to improve learning or decision-making in RL. This method proves valuable when raw sensory data is insufficient to obtain an optimal policy or additional domain knowledge can expedite learning. RL with side information can be divided into two categories: (1) RL with human input, where the agent receives explicit feedback or guidance from a human user, and (2) RL with auxiliary tasks, where the agent learns a related but more straightforward task to improve learning on the main task [45].

### 3.3. RL algorithms

Several algorithms can be used for RL. Some of the most common algorithms are:

- **Q-learning:** This value-based RL algorithm learns a policy for selecting actions by estimating the expected long-term reward for each action [46].
- **SARSA:** This value-based RL algorithm learns a policy by estimating the expected reward for each action based on the current state and the next action taken [47].
- **DQN:** This deep RL algorithm uses a neural network to approximate the action-value function and learn a policy for selecting actions [30].
- **A2C:** This actor-critic RL algorithm uses a neural network to approximate the action-value function and learn a policy for selecting actions [48].
- **PPO:** This policy gradient RL algorithm uses a neural network to learn a policy for selecting actions by directly optimizing the policy objective [49].
- **TRPO:** This trust region policy optimization RL algorithm uses a neural network to learn a policy for selecting actions by optimizing the policy objective within a trust region [50].
- **DDPG:** This deep deterministic policy gradient RL algorithm uses a neural network to learn a deterministic policy for selecting actions [39].
- **TD3:** This variant of the DDPG algorithm uses two neural networks to learn a deterministic policy for selecting actions [51].
- **A3C:** This asynchronous actor-critic RL algorithm uses multiple parallel agents to learn a policy for selecting actions [48].
- **REINFORCE:** This is a Monte Carlo policy gradient RL algorithm that learns a policy for selecting actions by sampling trajectories and estimating the gradient of the expected reward concerning the policy parameters [52].
- **DPG:** This is a deterministic policy gradient RL algorithm that learns a deterministic policy for selecting actions by estimating the gradient of the expected reward for the policy parameters [53].
- **NAF:** This is a natural actor-critic RL algorithm that uses a neural network to learn a continuous action policy by estimating the gradient of the expected reward concerning the policy parameters [54].

- **CEM:** This cross-entropy method RL algorithm learns a policy by sampling actions from a parametrized distribution and adjusting the parameters to maximize the expected reward [55].
- **ES:** This is an evolutionary strategy RL algorithm that learns a policy by evolving the parameters of a parametrized distribution through selection and mutation [56].
- **EKF-based RL:** This RL algorithm uses an extended Kalman filter to estimate the state-action value function and learn a policy for selecting actions [57].
- **Fitted Q-iteration:** This batch RL algorithm learns a policy by fitting a model of the action-value function to a dataset of transitions and using the fitted model to improve the policy iteratively [58].
- **GPT-based RL:** This RL algorithm uses a transformer-based language model (such as GPT) to learn a policy for selecting actions [59].

## 4. Previous Works That Used Rl For The Evrp

In [60], the authors consider the problem of EV routing with constraints on loading capacity, time window, and vehicle-to-grid (V2G) energy supply (CEVRPTW-D), which not only satisfies multiple system objectives but also scales efficiently to large problem sizes involving hundreds of customers and discharge stations. They introduce Quick Route Finder, which leverages RL for EV routing to address these issues. Using Solomon datasets [61], outcomes from RL are contrasted to exact formulations based on the mixed-integer linear program (MILP) and genetic algorithm (GA) metaheuristics. On average, the findings reveal that RL is 24 times quicker than MILP and GA while being near in quality (within 20%) to the ideal. They are continually working on creating improved RL models to reduce this optimality gap and manage to change demands in real-time.

In [62], the authors of this study employed a hyper-heuristic (HH) technique called Hyper-heuristic Adaptive Simulated Annealing with Reinforcement Learning (HHASARL) suggested. It incorporates a multi-armed bandit approach and the self-adaptive Simulated Annealing (SA) metaheuristic algorithm for addressing the CEVRP issue. Due to the restricted number of charging stations and the trip range of EVs, the EVs must need battery recharging moments in advance and cut travel times and expenses. The HH implemented improves numerous minimal best-known solutions and provides the best mean values for various high-dimensional examples for the proposed benchmark for the IEEE WCCI2020 competition [63]. In future studies, they want to alter the internal Adjust Station block of the suggested HH with a new way to make predicting pauses at the charging stations more efficient. In addition, deep RL approaches will replace the RL block to make it more resilient and evaluate the adaptability and efficiency of applying these techniques as a heuristic selection mechanism.

In [64], the authors create an RL-based algorithm to manage a community-owned electric vehicle fleet that offers ride-hailing services to local citizens. The electric vehicle fleet aims to reduce customer waiting time, power cost, and operational expenses of the cars. A new system defined by decentralized learning and centralized decision-making is offered to tackle the electric

vehicle fleet dispatch challenge. The decentralized learning approach enables the individual cars to share their operational experiences and deep neural network model for state-value function estimation, which mitigates the curse of dimensionality of state and action domains. The centralized decision-making framework reduces the vehicle fleet coordination issue into a linear assignment problem with polynomial time complexity. Numerical research findings reveal that the suggested technique beats the benchmark algorithms regarding societal cost reduction.

In [65], the authors suggest a RL model to manage power supply and demand uncertainties by deploying a collection of electric cars to deliver energy to various customers at different places. An electric vehicle is installed with multiple energy resources (e.g., PV panel, energy storage) that share power-producing units and storage among different users to power their premises to lower energy expenses. The performance of the RL model is examined under several configurations of customers and electric cars, compared to the findings from CPLEX and three heuristic techniques. The simulation findings reveal that the RL algorithm may cut energy costs up to 22.05%, 22.57%, and 19.33% compared to the genetic algorithm, particle swarm optimization, and artificial fish swarm algorithm results.

In [66], The authors explore the issue of vehicle routing for an EV fleet with V2G and battery swapping (SWP). The restrictions encompass loading capacity and delivery time constraints, intending to reduce the overall delivery costs. They enhance previous methodologies with a learning agent (LA) that grows to enormous issue sizes, including hundreds of clients, discharge stations, and battery-switching sites. Using two typical datasets (Solomon and Homberger) and a postal delivery network from Bangalore, they test LA against a GA and three different baselines. Their experimental assessment demonstrates that LA is 5.65 times quicker than GA. At the same time, GA is more accurate than LA. The LA can scale to problem cases with 400+ nodes, but the GA can only scale up to 200 nodes.

In [67], they offer an efficient Deep RL-based approach for constraint-based routing while considering electric vehicles' charging policies concurrently. They build a two-layer model to identify near-optimal solutions and manage various issue cases according to the rewards. The predefined feasibility time-consuming first layer approximates a series of successive actions in reality, providing the least time-consuming viable path without re-training for each new issue occurrence. The second step is to build a charging scheme along the previously defined viable route. The suggested technique is independent of the road network layout and the electric vehicles' kinds. Besides, the convergence of value function in the provided model for EVRP is explored. The experiment reveals that their technique exceeds the old ones in computing time with equivalent solution quality. Moreover, the developed model may be immediately utilized to tackle different issue cases on diverse road networks without re-training processes.

In [68], they study a cost optimization issue for plug-in hybrid electric vehicles (PHEVs) utilized for service delivery in the context of energy consumption unpredictability. For the cost optimization issue, an optimum strategy is discovered that dynamically chooses, as the vehicle drives, the car should be charged at which charging station to reduce the service fuel cost. The

issue is stated as a Partially Observable Markov Decision Process (POMDP) and is addressed using RL. The RL charging policy (RLCP), discovered by solving the POMDP, is compared to two benchmark policies and reveals that RLCP outperforms both. Most crucially, RLCP may be automatically adjusted to large fluctuations in the vehicle's energy consumption behavior by constantly training the RLCP model according to the most current information collected from the vehicle's environment.

In [69], utilizing the dataset presented in [10], the authors built an RL framework for solving the EVRPTW. Although the answers obtained for tiny cases by the suggested method could be better, they feel it is extremely promising. The reasons are three-fold: foremost, the algorithm displays remarkable scalability. It can solve cases of gigantic sizes which are unsolvable with any known approaches. Their investigation demonstrates that the suggested model may swiftly capture crucial information buried in the graph and deliver reasonably excellent possible solutions. Though not optimum, such practical methods might be leveraged to enable large-scale real-time EV operations. Secondly, the suggested model is particularly efficient in solving the EVRPTW. Numerous graph components, such as customers' needs, time frames, and the availability of charging services, might alter immediately. The RL model's capacity to effectively solve the issue enables the EV operators to immediately make modifications to face the problems stemming from the stochastic nature of the EVPRTW. Thirdly, the suggested model may be expanded to various variations of the EVRPTW. Practitioners can extend the proposed method by slightly tailoring the masking schemes and the reward function according to their operational constraints and objectives. It is much easier than adjusting other exact or metaheuristic algorithms that usually require certain assumptions and domain knowledge. Theoretically, the suggested solution strategy integrates the embedding graph techniques with the PN design, enabling the algorithm to synthesize the local and global knowledge to solve the target issue. They think its applicability is not limited to addressing EVRPTW as it might suit other CO issues that involve both local and global states of the graph on which it is defined.

In [18], the DS-EVRP was introduced and conceptualized as a Markov Decision Process. The study presents a solution technique rooted in Safe RL, featuring the subsequent contributions: firstly, a Value Function Approximation (VFA) utilizing a streamlined state representation to reduce Q-table dimensions and enhance exploration; secondly, a chance-constrained policy with dual safety levels aimed at curbing energy consumption and averting failures (specifically, battery depletion during transit); and thirdly, a training strategy employing tabu search based on heuristics to enhance exploration. Additionally, a series of computer experiments are conducted to assess the proposed solution approach and scrutinize its characteristics. Compared to the deterministic online optimization strategy, their technique has the potential to achieve energy savings of 4.8% (up to 12%) through anticipatory route planning and charging. The proposed training approach demonstrates promising efficacy even when dealing with a limited number of episodes, attributed to the effective utilization of the rollout function and the VFA technique.

## 5. Conclusion And Future Work

RL approaches have shown excellent potential for solving several optimization problems, Including VRP, EVRP, and their variants. We covered several studies that have applied RL to the EVRP and have demonstrated its effectiveness in finding optimal routes for a fleet of EVs. One of the main challenges of using RL for EVRP is the complexity of the problem. EVRP involves many variables and constraints, such as vehicle capacity, battery capacity, charging infrastructure, and customer demand. Accurately modeling these factors can be challenging, and finding an optimal solution within a reasonable timeframe can be even more demanding. Another challenge is dealing with uncertainty. In real-world applications, it can be difficult to predict customer demand or the availability of charging infrastructure with certainty. RL algorithms based on deterministic models may need to be better-suited to dealing with these uncertain environments. Conversely, an inherent advantage of applying RL to EVRP lies in its capacity to dynamically adapt to evolving conditions. RL algorithms can learn from experience and update decision-making processes as new information becomes available. This ability allows RL algorithms to adjust to changing customer demand or new charging infrastructure as they become available. Another noteworthy opportunity is scalability. RL methods can solve problems with vast dimensionality and complexity, making it possible to optimize for more significant and realistic instances of EVRP. Further research is needed to develop more efficient RL algorithms for solving the EVRP and better understand the following:

- **Exploration and exploitation trade-off:** EVRP involves balancing the exploration of different routes with the exploitation of the best-known routes. Research could focus on developing RL algorithms that find a balance between the two by exploring new routes while efficiently exploiting the best-known routes.
- **Combining RL with other techniques:** Integrating RL with additional optimization methodologies, such as mathematical programming or evolutionary algorithms, has the potential to yield more efficient solutions.
- **Real-time decision-making:** EVRP constitutes a dynamic problem, characterized by evolving customer demands and fluctuating charging infrastructure over time. Research endeavors could be channeled towards developing real-time RL algorithms capable of making instantaneous decisions, incorporating the latest information about customer demand and charging infrastructure.
- **Incorporating physical constraints:** EVRP entails numerous physical constraints such as battery capacity and vehicle range. Employing RL to integrate these constraints into the decision-making process could facilitate discovering solutions that align more closely with reality.
- **Multi-agent systems:** Another avenue for exploration lies in multi-agent RL algorithms, wherein distinct agents govern multiple vehicles and charging stations, necessitating coordinated actions. This approach promises increased realism and the potential to derive more efficient solutions.
- **Real-world deployment:** Despite the strides taken in simulating and testing RL-based EVRP, a critical imperative exists for further research that centers on the practical deployment of these

algorithms in real-world contexts. Such endeavors should encompass the evaluation of performance and scalability, grounded in actual data, to engender a comprehensive understanding of their viability and effectiveness.

**References**

[1] A. Sbihi and R. W. Eglese, "Combinatorial optimization and green logistics," 4OR, vol. 5, no. 2, pp. 99–116, 2007.

[2] A. Afroditi, M. Boile, S. Theofanis, E. Sdoukopoulos, and D. Margaritis, "Electric vehicle routing problem with industry constraints: trends and insights for future research," Transportation Research Procedia, vol. 3, pp. 452–459, 2014.

[3] S. Shao, W. Guan, B. Ran, Z. He, and J. Bi, "Electric vehicle routing problem with charging time and variable travel time," Mathematical Problems in Engineering, vol. 2017, 2017.

[4] D. Margaritis, A. Anagnostopoulou, A. Tromaras, and M. Boile, "Electric commercial vehicles: Practical perspectives and future research directions," Research in Transportation Business & Management, vol. 18, pp. 4– 10, 2016.

[5] A. Felipe, M. T. Ortu ´ no, G. Righini, and G. Tirado, "A ˜ heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," Transportation Research Part E: Logistics and Transportation Review, vol. 71, pp. 111–128, 2014.

[6] J. Lin, W. Zhou, and O. Wolfson, "Electric vehicle routing problem," Transportation research procedia, vol. 12, pp. 508–521, 2016.

[7] N. Bilisbekov, C. Sarfo, A. Dandis, and M. Eid, "Linking bank advertising to customer attitudes: The role of cognitive and affective trust," Management science letters, vol. 11, no. 4, pp. 1083–1092, 2021.

[8] M. Bruglieri, F. Pezzella, O. Pisacane, and S. Suraci, "A variable neighborhood search branching for the electric vehicle routing problem with time windows," Electronic Notes in Discrete Mathematics, vol. 47, pp. 221–228, 2015.

[9] O. Sassi, W. R. Cherif-Khettaf, and A. Oulamara, "Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles," in Modelling, computation and optimization in information systems and management sciences. Springer, 2015, pp. 57–68.

[10] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," Transportation science, vol. 48, no. 4, pp. 500–520, 2014.

[11] A. Montoya, C. Gueret, J. E. Mendoza, and J. Villegas, ´ "The electric vehicle routing problem with partial charging and nonlinear charging function," Ph.D. dissertation, LARIS, 2015.

[12] D. Goeke and M. Schneider, "Routing a mixed fleet of electric and conventional vehicles," European Journal of Operational Research, vol. 245, no. 1, pp. 81–99, 2015.

[13] R. Roberti and M. Wen, "The electric traveling salesman problem with time windows," Transportation Research Part E: Logistics and Transportation Review, vol. 89, pp. 32–52, 2016.

[14] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," Computers & Industrial Engineering, vol. 161, p. 107650, 2021.

[15] Y.-H. Jia, Y. Mei, and M. Zhang, "A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem," IEEE Transactions on Cybernetics, 2021.

[16] J. Lu, Y. Chen, J.-K. Hao, and R. He, "The timedependent electric vehicle routing problem: Model and solution," Expert Systems with Applications, vol. 161, p. 113593, 2020.

[17] Y. Zhang, S. Zhou, X. Ji, B. Chen, H. Liu, Y. Xiao, and W. Chang, "The mathematical model and an genetic algorithm for the two-echelon electric vehicle routing problem," in Journal of Physics: Conference Series, vol. 1813, no. 1. IOP Publishing, 2021, p. 012006.

[18] R. Basso, B. Kulcsar, I. Sanchez-Diaz, and X. Qu, ´ "Dynamic stochastic electric vehicle routing with safe reinforcement learning," Transportation research part E: logistics and transportation review, vol. 157, p. 102496, 2022.

[19] N. Wang, Y. Sun, and H. Wang, "An adaptive memetic algorithm for dynamic electric vehicle routing problem with time-varying demands," Mathematical Problems in Engineering, vol. 2021, 2021.

[20] B. Zhou and Z. Zhao, "Multi-objective optimization of electric vehicle routing problem with battery swap and mixed time windows," Neural Computing and Applications, pp. 1–24, 2022.

[21] Q.-q. Yang, D.-w. Hu, H.-f. Chu, and C.-r. Xu, "An electric vehicle routing problem with pickup and delivery," in CICTP 2018: Intelligence, Connectivity, and Mobility. American Society of Civil Engineers Reston, VA, 2018, pp. 176–184.

[22] M. E. H. Sadati, V. Akbari, and B. C¸ atay, "Electric vehicle routing problem with flexible deliveries," International Journal of Production Research, pp. 1–27, 2022.

[23] X. Zuo, Y. Xiao, M. You, I. Kaku, and Y. Xu, "A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function," Journal of Cleaner Production, vol. 236, p. 117687, 2019.

[24] Z. Wu and J. Zhang, "A branch-and-price algorithm for two-echelon electric vehicle routing problem," Complex & Intelligent Systems, pp. 1–16, 2021.

[25] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.

[27] R. S. Sutton, A. G. Barto et al., "Introduction to reinforcement learning," 1998.

[28] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," Machine learning, vol. 22, no. 1, pp. 33–57, 1996.

[29] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," Artificial intelligence, vol. 72, no. 1-2, pp. 81–138, 1995.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529–533, 2015.

[31] K. Zhang, Z. Yang, and T. Bas¸ar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," Handbook of Reinforcement Learning and Control, pp. 321–384, 2021.

[32] R. S. Sutton, "Learning to predict by the methods of temporal differences," Machine learning, vol. 3, no. 1, pp. 9–44, 1988.
[33] T. G. Dietterich, "Hierarchical reinforcement learning with the maxq value function decomposition," Journal of artificial intelligence research, vol. 13, pp. 227–303, 2000.
[34] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey." Journal of Machine Learning Research, vol. 10, no. 7, 2009.
[35] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 1.
[36] J. Ho and S. Ermon, "Generative adversarial imitation learning," Advances in neural information processing systems, vol. 29, 2016.
[37] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," Advances in neural information processing systems, vol. 29, 2016.
[38] M. P. Deisenroth, G. Neumann, J. Peters et al., "A survey on policy search for robotics," Foundations and Trends® in Robotics, vol. 2, no. 1–2, pp. 1–142, 2013.
[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
[40] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," Journal of artificial intelligence research, vol. 4, pp. 237–285, 1996.
[41] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile ´ sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in Distributed autonomous robotic systems 5. Springer, 2002, pp. 299–308.
[42] J. Garcıa and F. Fernandez, "A comprehensive survey ´ on safe reinforcement learning," Journal of Machine Learning Research, vol. 16, no. 1, pp. 1437–1480, 2015.
[43] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in International conference on machine learning. PMLR, 2017, pp. 1146–1155.
[44] M. M. Drugan, "Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms," Swarm and evolutionary computation, vol. 44, pp. 228–246, 2019.
[45] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional nonconvex optimization," Advances in neural information processing systems, vol. 27, 2014.
[46] C. J. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, no. 3, pp. 279–292, 1992.
[47] G. A. Rummery and M. Niranjan, On-line Q-learning using connectionist systems. University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.
[48] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in International conference on machine learning. PMLR, 2016, pp. 1928–1937.

[49] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
[50] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in International conference on machine learning. PMLR, 2015, pp. 1889–1897.
[51] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in International conference on machine learning. PMLR, 2018, pp. 1587–1596.
[52] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," Machine learning, vol. 8, no. 3, pp. 229–256, 1992.
[53] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in International conference on machine learning. PMLR, 2014, pp. 387–395.
[54] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in International conference on machine learning. PMLR, 2016, pp. 2829–2838.
[55] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," Methodology and computing in applied probability, vol. 1, no. 2, pp. 127–190, 1999.
[56] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," arXiv preprint arXiv:1703.03864, 2017.
[57] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.
[58] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," Journal of Machine Learning Research, vol. 6, 2005.
[59] V. Uc-Cetina, N. Navarro-Guerrero, A. Martin-Gonzalez, C. Weber, and S. Wermter, "Survey on reinforcement learning for language processing," Artificial Intelligence Review, pp. 1–33, 2022.
[60] A. Narayanan, P. Misra, A. Ojha, V. Bandhu, S. Ghosh, and A. Vasan, "A reinforcement learning approach for electric vehicle routing problem with vehicle-to-grid supply," arXiv preprint arXiv:2204.05545, 2022.
[61] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," Operations research, vol. 35, no. 2, pp. 254–265, 1987.
[62] E. Rodr´ıguez-Esparza, A. D. Masegosa, D. Oliva, and E. Onieva, "A new hyper-heuristic based on adaptive simulated annealing and reinforcement learning for the capacitated electric vehicle routing problem," arXiv preprint arXiv:2206.03185, 2022.
[63] J. Zhao, M. Mao, X. Zhao, and J. Zou, "A hybrid of deep reinforcement learning and local search for the vehicle routing problems," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 11, pp. 7208–7218, 2020.
[64] J. Shi, Y. Gao, W. Wang, N. Yu, and P. A. Ioannou, "Operating electric vehicle fleet for ride-hailing services with reinforcement learning," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 11, pp. 4822–4834, 2019.

[65] M. Alqahtani and M. Hu, “Dynamic energy scheduling and routing of multiple electric vehicles using deep reinforcement learning,” Energy, vol. 244.

[66] A. Narayanan, P. Misra, A. Ojha, A. Gupta, S. Ghosh, and A. Vasan, “Agent-based learning approach to electric vehicle routing problem with vehicle-to-grid supply and battery swapping,” in Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD), 2023, pp. 185–193.

[67] Y. Zhang, M. Li, Y. Chen, Y.-Y. Chiang, and Y. Hua, “A constraint-based routing and charging methodology for battery electric vehicles with deep reinforcement learning,” IEEE Transactions on Smart Grid, 2022.

[68] T. Panayiotou, S. P. Chatzis, C. Panayiotou, and G. Ellinas, “Charging policies for phevs used for service delivery: a reinforcement learning approach,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 1514– 1521.

[69] B. Lin, B. Ghaddar, and J. Nathwani, “Deep reinforcement learning for the electric vehicle routing problem with time windows,” IEEE Transactions on Intelligent Transportation Systems, 2021.