batteries

MDPI

Article

Using Reinforcement Learning to Solve a Dynamic Orienteering Problem with Random Rewards Affected by the Battery Status

Angel A. Juan [1,*], Carolina A. Marugan [1], Yusef Ahsini [1], Rafael Fornes [1], Javier Panadero [2] and Xabier A. Martin [1]

[1] Research Center on Production Management and Engineering, Universitat Politècnica de València, Plaza Ferrandiz-Salvador, 03801 Alcoy, Spain
[2] Department of Computer Architecture & Operating Systems, Universitat Autònoma de Barcelona, Carrer de les Sitges, 08193 Bellaterra, Spain
* Correspondence: ajuanp@upv.es

Abstract: This paper discusses an orienteering optimization problem where a vehicle using electric batteries must travel from an origin depot to a destination depot while maximizing the total reward collected along its route. The vehicle must cross several consecutive regions, with each region containing different types of charging nodes. A charging node has to be selected in each region, and the reward for visiting each node—in terms of a 'satisfactory' charging process—is a binary random variable that depends upon dynamic factors such as the type of charging node, weather conditions, congestion, battery status, etc. To learn how to efficiently operate in this dynamic environment, a hybrid methodology combining simulation with reinforcement learning is proposed. The reinforcement learning component is able to make informed decisions at each stage, while the simulation component is employed to validate the learning process. The computational experiments show how the proposed methodology is capable of design routing plans that are significantly better than non-informed decisions, thus allowing for an efficient management of the vehicle's battery under such dynamic conditions.

Keywords: orienteering problem; battery management; electric vehicle; reinforcement learning; simulation

Citation: Juan, A.A.; Marugan, C.A.; Ahsini, Y.; Fornes, R.; Panadero, J.; Martin, X.A. Using Reinforcement Learning to Solve a Dynamic Orienteering Problem with Random Rewards Affected by the Battery Status. *Batteries* **2023**, *9*, 416. https://doi.org/10.3390/batteries9080416

Academic Editor: Prodip K. Das

Received: 19 July 2023
Revised: 4 August 2023
Accepted: 8 August 2023
Published: 9 August 2023

1. Introduction

Modern transportation and logistics systems call for efficient vehicle routing solutions that can adapt to dynamic and uncertain conditions. This is particularly relevant for electric vehicles (EVs), which are gaining popularity due to their potential for reducing transportation costs and minimizing environmental impact. The increasing adoption of EVs has sparked a revolution in the automotive industry, offering a sustainable alternative to conventional internal combustion engine vehicles [1]. However, these vehicles face several challenges when it comes to real-time route planning under dynamic scenarios, such as the need to consider weather conditions, traffic or on-site congestion, and travel time. In addition, as EVs rely heavily on batteries for their power supply, effective battery management is crucial to ensure optimal performance, an extended battery life, and enhanced driving range. Moreover, the need for an efficient battery management system (BMS) calls for a cutting-edge technology combined with a more refined software. Any BMS has to consider the uncertainty associated with the availability of charging stations along the route. Finding reliable charging infrastructure can be a daunting task, as charging stations are still relatively scarce in many areas. This lack of widespread charging infrastructure poses a significant obstacle for EV owners, as it restricts the number of feasible options.

For long-distance routes, there comes a time when the vehicle needs to face the problem of choosing the next charging station point. This can be challenging given the

environmental dynamic conditions mentioned above. The distinction becomes crucial, especially for aged batteries, when considering charging up having a 60% charge level versus charging up having a 20% level. Therefore, an 'optimal' route towards the charging station point should be taken into account, so that given the distance from the current location to the station, the congestion and weather conditions, and the available chargers, the goal is to maximize the lifespan of the batteries while ensuring a charging time that is acceptable, as discussed in Abdollahi et al. [2].

Initially proposed by Golden et al. [3], the orienteering problem (OP) is named after the sport of orienteering, where a runner selects specific points to navigate between within a given time limit. The OP involves a vehicle that begins at a depot, traverses a set of nodes, and finally reaches a destination depot, with the objective of maximizing the total reward earned from visiting the nodes. The problem becomes more challenging as the vehicle has a limited distance or time to cover the path from the origin node to the destination node. The OP and its extension to multiple vehicles, the team orienteering problem, have found relevance in modern applications such as unmanned aerial vehicles and road EVs with limited battery capacity [4,5]. This paper explores a dynamic OP where a single vehicle must travel from an origin to a destination while visiting specific charging nodes along the way to ensure it can reach its final destination (Figure 1). The vehicle needs to pass through multiple regions and, since a region represents a set of clustered charging stations, it makes sense to assume that it will only visit one charging station per region. Once charged at a station, it can then resume its travel to the next region until it finally reaches the destination node. Each region contains different types of charging nodes, e.g., alternating current (AC) level 2, direct current (DC) fast, superchargers, etc. The reward associated with visiting each charging node is represented by a binary random variable, indicating whether or not the vehicle arrived at the charging node within the 'optimal' charging window to prolong the battery's lifespan and to complete the charging process in a reasonable time period. In order to consider realistic context conditions, it is assumed that the probability of receiving a positive reward at each charging node depends on dynamic factors such as: (i) weather conditions along the route to the charger; (ii) congestion on the road or at the charging station; (iii) travel time from the vehicle's current position to the charger's location; and (iv) the age of the battery.

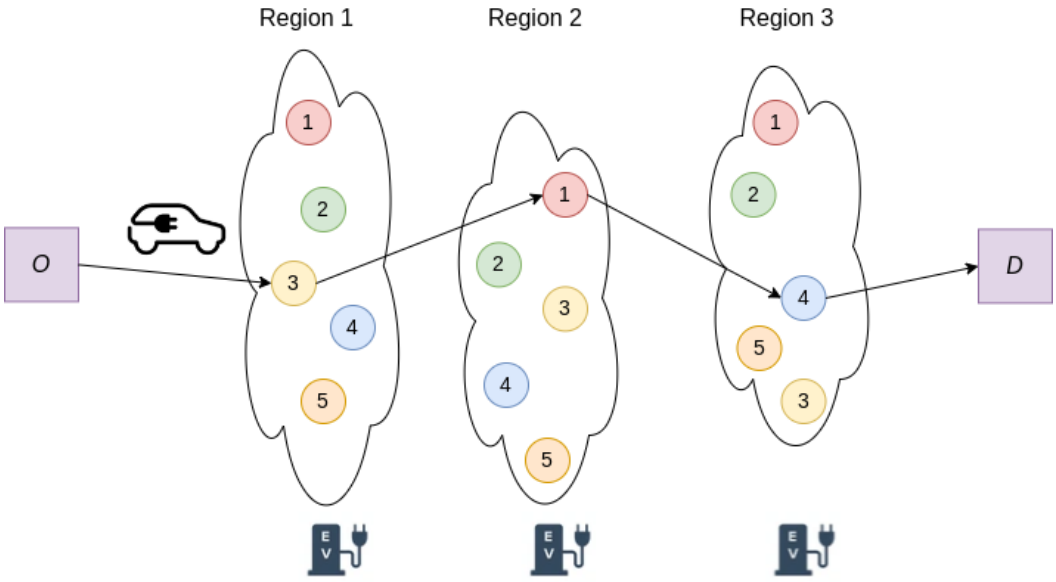

Figure 1. An orienteering problem with 3 regions and 5 potential charging nodes per region.

To address the aforementioned dynamic orienteering problem, this paper proposes the combination of simulation with reinforcement learning (RL) [6] to emulate how the

system would perform in a real-life scenario. Our approach involves training a machine learning model, which can be a slow process in a real-life scenario. Hence, to illustrate how this learning process works, a simulation experiment is proposed. The simulation allows us to emulate how the learning process would occur in a real-life scenario and test the quality of the learning models employed in our approach. Thus, the main contribution of this paper is the combination of reinforcement learning with simulation to illustrate how to make informed decisions in a dynamic OP with electric vehicles and binary random rewards that can be influenced by several factors, including their battery status.

The remainder of the paper is structured as follows: Section 2 gives a more extensive context on the types of EVs and their integration into the ongoing transportation network, current research in the field, battery capacity, materials and charging time, and several factors that might influence the state of the battery. Section 3 provides a brief review of OPs. Section 4 describes the problem in a more formal way. Section 5 introduces the numerical case study that is considered. Section 6 discusses the details of the simulation experiment as well as the reinforcement learning algorithm employed. Section 7 analyzes the results obtained in our computational test. Finally, Section 8 highlights the main conclusions of our work and suggests open research lines.

2. Some Technical Details on EVs

The transportation sector accounted for 23% of energy-related emissions in 2019 [7]. The largest source of transport emissions are road vehicles, predominantly powered by internal combustion engines. Since EVs are a promising mitigation technology to reduce the impact on climate change, air pollutants, and noise [8], researchers have shown significant interest in studying EV energy consumption due to its importance for understanding the efficiency, performance, and environmental impact [9,10]. Several models can be used to calculate energy consumption in EVs, and Qi et al. [11] classified them into three main categories: analytical, statistical, and computational models. While accurate estimation and prediction of EVs' range is crucial for driver confidence and planning, the design and optimization of charging infrastructure networks also play an important role in ensuring efficient charging processes and minimizing energy consumption. There are several types of EVs available on the market [12]. These range from conventional vehicles with no electric components to fully EVs, including hybrid electric vehicles (HEVs) as well as battery electric vehicles (BEVs). Each type has its own advantages and considerations, and the choice depends on factors such as driving habits, range requirements, the availability of charging infrastructure, and personal preferences.

As discussed in [13], BEVs produce zero tailpipe emissions but require charging infrastructure to recharge their batteries. The battery is the core component of an EV, especially for BEVs, and can be categorized by qualities such as its specific energy, specific power, capacity, voltage, and chemistry [14]. Battery modeling is quite complex and many models have been developed: empirical, equivalent circuit, physics-based, and high level stochastic [15]. The initial technology used was a lead–acid battery that was soon replaced because of its drawbacks such as having low energy density and being heavy. Nickel-based batteries were introduced soon after as a mature technology, but they have longer recharging times and poor performance in cold weather. Currently, lithium-based batteries are the most common and widely adopted in the EV industry [16]. Even though some methods have been proposed to predict the remaining useful life of lithium batteries [17], there are still several open issues with their use: in particular, their safety as well as environmental impact concerns.

One of the main challenges raised by BEVs is to obtain acceptable driving ranges. The state of charge (SOC) measurement is particularly important as it indicates the maximum driving range. The capacity of EV batteries can range from around 20 kWh in smaller electric cars to over 100 kWh in larger, high-end electric vehicles. However, accurately estimating driving distance is difficult, and factors such as the vehicle's efficiency, driving style, and weather conditions also influence the actual driving range achieved [18,19]. The

optimal SOC range in batteries can vary depending on the specific type of lithium-based battery chemistry and the desired operating parameters [20]. According to Koltermann et al. [21], the capabilities are limited at the border areas of the SOC. The results show that batteries can only safely deliver full power without a detrimental impact on their health and longevity at between 20–80% of the SOC. At both ends, the battery exhibits much higher polarization impedance [22]. The maximum SOC is usually set to ensure safe operation and prevent overcharging, stress on the material, or elevated operating temperatures [23]. Likewise, the minimum SOC tries to avoid an unexpected system shutdown or loss of power that can occur due to low voltage levels and the risk of cell imbalance. In addition, maintaining a low SOC can potentially contribute to driver anxiety. Advancements in charging infrastructure are also reducing concerns about range anxiety by providing faster and more accessible charging options for EV owners [24]. However, the number of charging stations is still relatively limited, so predicting the best place to charge the vehicle in dynamic conditions is important. There are several charging technologies available for EVs depending on the different power levels an EV can be charged to [25]. Level 1 charging (120 V AC) uses a household electrical outlet and provides around 1.4 to 1.9 kW. It provides the slowest charging rate for EVs. Level 2 charging (240 V AC) requires a compatible charging station or wall-mounted charger. It delivers power ranging from 3.3 kW to 22 kW. Level 3 or DC fast charging stations can provide charging powers ranging from 50 kW to over 350 kW. The vehicle's onboard charger and charging port specifications will determine the maximum charging rate it can accept. Among other factors, a vehicle's charge time depends on the level of the charger and the type of EV. According to the U.S. Department of Transportation [26], to reach an 80% battery level from empty can take between 40 and 50 h for a BEV, and between 5 and 6 h for a plug-in HEV (with a Level 1 charger). Likewise, it can take between 4 and 10 h for a BEV and between 1 and 2 h for a plug-in HEV (with a Level 2 charger). Finally, it can take between 20 min and 1 h for a BEV (with a Level 3 charger).

EV penetration causes significant issues on the power distribution grid, such as an increase in power demand, system losses, voltage drops, equipment overloading, and stability impact [27]. At the same time, some exciting opportunities appear with EV deployment on the smart grid, such as grid flexibility through vehicle-to-grid (V2G) technology, demand response, and the integration of renewable energy sources [25]. V2G mode allows EVs to discharge power back to the grid. This can support grid balancing, frequency regulation, and voltage stabilization. The weather conditions directly affect the battery temperature [28], the climate control [29], and the charging efficiency [30]. Extreme weather conditions can affect the temperature of the EV's battery. High temperatures can increase the risk of the battery overheating, which may reduce its performance and lifespan. Likewise, very cold temperatures can decrease the battery's efficiency and capacity temporarily. EVs often rely on climate control systems to maintain a comfortable cabin temperature. This includes features such as air conditioning in hot weather and heating in cold weather. The use of these systems can impact the overall energy consumption and range of the vehicle. Extreme temperatures can also affect the efficiency of charging systems. Congestion can generate an increase in the stop-and-go driving style. This can lead to more energy-intensive acceleration and braking, and thus result in increased energy consumption and a higher load on the battery, potentially reducing its overall range [31]. In addition, when stuck in congestion, EVs may be required to idle for extended periods, especially in situations where traffic is at a standstill. Idling consumes energy from the battery to power auxiliary systems such as climate control and entertainment systems. Prolonged idling can drain the battery charge faster and reduce the available range [32].

3. Related Work on Vehicle Orienteering Problems

The orienteering problem was first introduced by Golden et al. [3], who proved it to be NP hard. Early research examined the deterministic version of the problem within the framework of vehicle routing, where one vehicle chooses the nodes to visit and the order of visits within

a defined time frame. In contrast, research on the stochastic OP is relatively recent. The first study to include stochasticity in the OP was conducted by Ilhan et al. [33], which assumed that only node rewards were stochastic. Other researchers, including Campbell et al. [34], Papapanagiotou et al. [35], Verbeeck et al. [36], or Evers et al. [37], focused on cases where service and travel times were stochastic, with service times typically incorporated into travel times. Several approaches have been employed to solve the stochastic orienteering problem, including a combination of branch and bound algorithms with local search [34] and local search simulations [38]. Further refinements were made to these methods, with Varakantham and Kumar [39] utilizing a sample average approximation technique to improve on the results of Lau et al. [38]. Additionally, Zhang et al. [40] expanded the method of Campbell et al. [34] to include time windows for arriving at nodes. Gama and Fernandes [41] introduced a solution to the orienteering problem with time windows, employing Pointer Network models trained through reinforcement learning. A comprehensive review of the orienteering problem and its variants can be found in Gunawan et al. [42]. Still, to the best of our knowledge, our work is the first one that proposes the combined use of RL and simulation to deal with a version of the problem with stochastic rewards that depend upon dynamic conditions. In our view, this dynamic OP with stochastic rewards has relevant applications to scenarios involving EVs that require an efficient management of their batteries.

4. Modeling the Dynamic OP with Binary Random Rewards

Let $G = (V, E)$ be a directed graph with node set V and edge set E. Node $O \in V$ is the origin node, while node $D \in V$ is the destination node. Each node $i \in V \setminus \{O, D\}$ has a reward r_i associated with it, which is a binary random variable that takes the value 1 with probability p_i and 0 with probability $1 - p_i$. Let us denote this by region R_k, with $k \in K = \{1, 2, \ldots, |K|\}$, a subset of $V \setminus \{O, D\}$. The problem is to find a path P that starts at O and ends at D, maximizes the expected total reward collected along the path, and visits at most one node in each region R_k. In our case, the reward is based on extending the battery's lifespan [21], as well as on completing the charging process within a certain time interval. Thus, the probability of obtaining a reward for a recharging node i depends on the type of node as well as on the current status at the region R_k. This status is determined by dynamic context conditions, i.e., $p_i = f(i, R_k)$ for an unknown (black-box) function f.

Let x_i be a binary decision variable that takes the value 1 if node i is visited along the path P, and 0 otherwise. Let e_{ij} be a binary decision variable that takes the value 1 if there is an arc from node i to node $j \neq i$ along the path P, and 0 otherwise. Then, the problem can be formulated as follows:

$$\max \sum_{i \in V \setminus \{O,D\}} r_i p_i x_i \tag{1}$$

subject to:

$$\sum_{i \in V \setminus \{O\}} e_{Oi} = 1 \tag{2}$$

$$\sum_{j \in V \setminus \{i,O\}} e_{ij} - \sum_{j \in V \setminus \{i,D\}} e_{ji} = 0 \quad \forall i \in V \setminus \{O, D\} \tag{3}$$

$$\sum_{j \in V \setminus \{D\}} e_{jD} = 1 \tag{4}$$

$$\sum_{i \in R_k} x_i = 1 \qquad \forall k \in K \tag{5}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{6}$$

$$e_{ij} \in \{0, 1\} \qquad \forall (i, j) \in E \tag{7}$$

The objective function Equation (1) maximizes the expected total reward collected along the path P. Constraint Equations (2) and (4) ensure that the path starts at the origin node O and ends at the destination node D, respectively. Constraint Equation (3) ensures that the flow of the path P is conserved at each node $i \in V \setminus \{O, D\}$, i.e., the number of incoming arcs is equal to the number of outgoing arcs. Constraint Equation (5) ensures that at most one node in each region R_k is visited. Constraint Equations (6) and (7) enforce the binary variables.

5. A Numerical Case Study

Consider a dynamic OP, with binary random demands $y \in \{0, 1\}$, similar to the one represented in Figure 1. Let us assume that a vehicle must now cross six consecutive regions, with each region containing five different types of charging nodes, each type with a different probability of obtaining a reward. In accordance with Section 2, let us assume that such probability will depend on a vector of factors that describe the context conditions at each region–node pair, which are: battery age, congestion, weather, battery status, and time of charge. The logistic sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ maps real-valued numbers to a range between 0 and 1. In particular, for each type of node $i \in \{1, 2, \ldots, 5\}$, let us assume that the real-life probability of obtaining a reward p_i is modeled as a logistic function f, which is defined for each region, k_{1i} to k_{6i} (Table 1), and each of the following factors: (i) battery age (cycles) $ba \in \{10, 1000\}$; (ii) current battery status (in %) $be_i \in \{0, 100\}$; (iii) congestion $c_i \in \{0, 1\}$ (where $c = 1$ represents high congestion and $c = 0$ represents low congestion); (iv) weather conditions $w_k \in \{0, 1\}$ (where $w = 1$ represents good weather and $w = 0$ represents bad weather); and (v) time of charge (in hours) $tc_i \in (0, 30)$. Hence, the real-life probability p_i of obtaining a reward when visiting a node of type i in region R_k under context conditions $(ba, be_i, c_i, w_k, tc_i)$ is given by:

$$p_i = \frac{1}{1 + \exp(-(k_{1i}w_k + k_{2i}c_i + k_{3i}tc_i + k_{4i}ba + k_{5i}(20 - |be_i - 20|) + k_{6i}))} \tag{8}$$

For each of the five considered nodes, Figure 2 shows the real probability of obtaining a reward of 1 when the node is visited under different combinations of congestion, weather conditions, battery age, and battery status. Notice also that charging time has a great influence on this probability for all nodes since the objective is to extend the lifetime of the battery without compromising the time at which the vehicle reaches its destination. Moreover, the node offering a higher probability of reward might vary according to the current weather, congestion, and other conditions in the region.

Table 1. Real-life parameters for each node type (coefficients of the associated logistic function).

Node Type	k_{1i}	k_{2i}	k_{3i}	k_{4i}	k_{5i}	k_{6i}
1	3.0	−0.5	−0.4	−0.0010	−0.00010	3.5
2	1.5	−0.1	−0.41	−0.0004	−0.00030	4.5
3	3.5	−0.1	−0.43	−0.0012	−0.00010	4.5
4	2.0	−0.2	−0.50	−0.0009	−0.00035	4.0
5	4.5	−0.9	−0.42	−0.0008	−0.00030	3.0

Despite the fact that the function f that represents the real probability of obtaining a reward has been properly defined, this will not usually be possible in a real-life application. In effect, the real-life parameters shown in Table 1 will be unknown in many practical applications. Then, given a vector x of five factors associated with a region–node combination, the goal will be to predict y, i.e., $\hat{y} = P(y = 1|x)$, so the next charging node i to be visited can be selected using this estimated probability. Hence, from this point on, it is assumed

that the true values of the parameters k_{1i} to k_{6i} are unknown, and a reinforcement learning algorithm is proposed in order to predict the real probability of obtaining a reward. In a real-life application, the algorithm achieves this predictive capacity by making decisions, observing the associated outcomes, and then learning from these interactions with reality. In online RL, the learning agent interacts directly with the environment in real-time. The agent makes decisions, receives feedback (rewards), and updates its policy based on the observed outcomes. The agent explores (tries out different options) and exploits (selects the best-known option) the environment simultaneously, making decisions and adapting its behavior as it interacts with the environment. In order to illustrate this learning process, simulation is employed to emulate this interaction between the algorithm and reality. A total of $10,000$ trips were simulated.

At each region R_k, the algorithm must select a node type i to visit. This selection is based on the current estimate of the expected reward for each node type i, and the ϵ parameter controls the balance between exploration and exploitation. Specifically, with probability $1-\epsilon$, the algorithm selects the node type with the highest expected reward (exploitation), and with probability ϵ, it selects a node type at random (exploration). After selecting a node type, the algorithm receives feedback from reality (the simulation in our case) in the form of a binary reward, and updates its estimate of the expected reward for each node type accordingly.

In the computational experiments a hybrid gradient boosting with decision trees model is utilized to estimate the expected reward for each region–node combination based on the feedback provided by the simulation environment. During the simulation, random values are generated to emulate the context conditions of each region. Then, using the black-box function f, the response that would be obtained in a real-life environment is estimated. The algorithm uses this response to iteratively enhance the model that predicts, for each node and contextual values, the probability of obtaining a reward. Thus, for example, suppose that after several simulation runs, the algorithm learns that nodes of type 1 are the most rewarding under good weather and low congestion conditions, nodes of type 2 are the most rewarding under bad weather and high congestion, nodes of type 3 are the most rewarding under good weather and high congestion, nodes of type 4 are the most rewarding under bad weather and low congestion, and nodes of type 5 are the most rewarding under any other context condition, it then adjusts its policy accordingly and continues to improve over time.

A conceptual schema summarizing the described methodology is provided in Figure 3. The simulation component provides new testing conditions for the RL component to make decisions (step 1). The RL component makes decisions by selecting the next charging node in the routing plan (step 2). Then, the black-box function emulating real life is employed to check the real impact of the decision suggested by the RL agent (step 3) and provide feedback to it (step 4). Finally, the simulation of a new scenario is activated and the process is repeated until a complete solution (a selection of charging nodes connecting the origin depot with the destination depot) is built. At this stage, the entire loop is iterated for a number of runs. As more and more runs are executed, the more trained the RL model becomes and, hence, the better its decisions are in terms of which charging node has to be visited next according to the vector of factors.

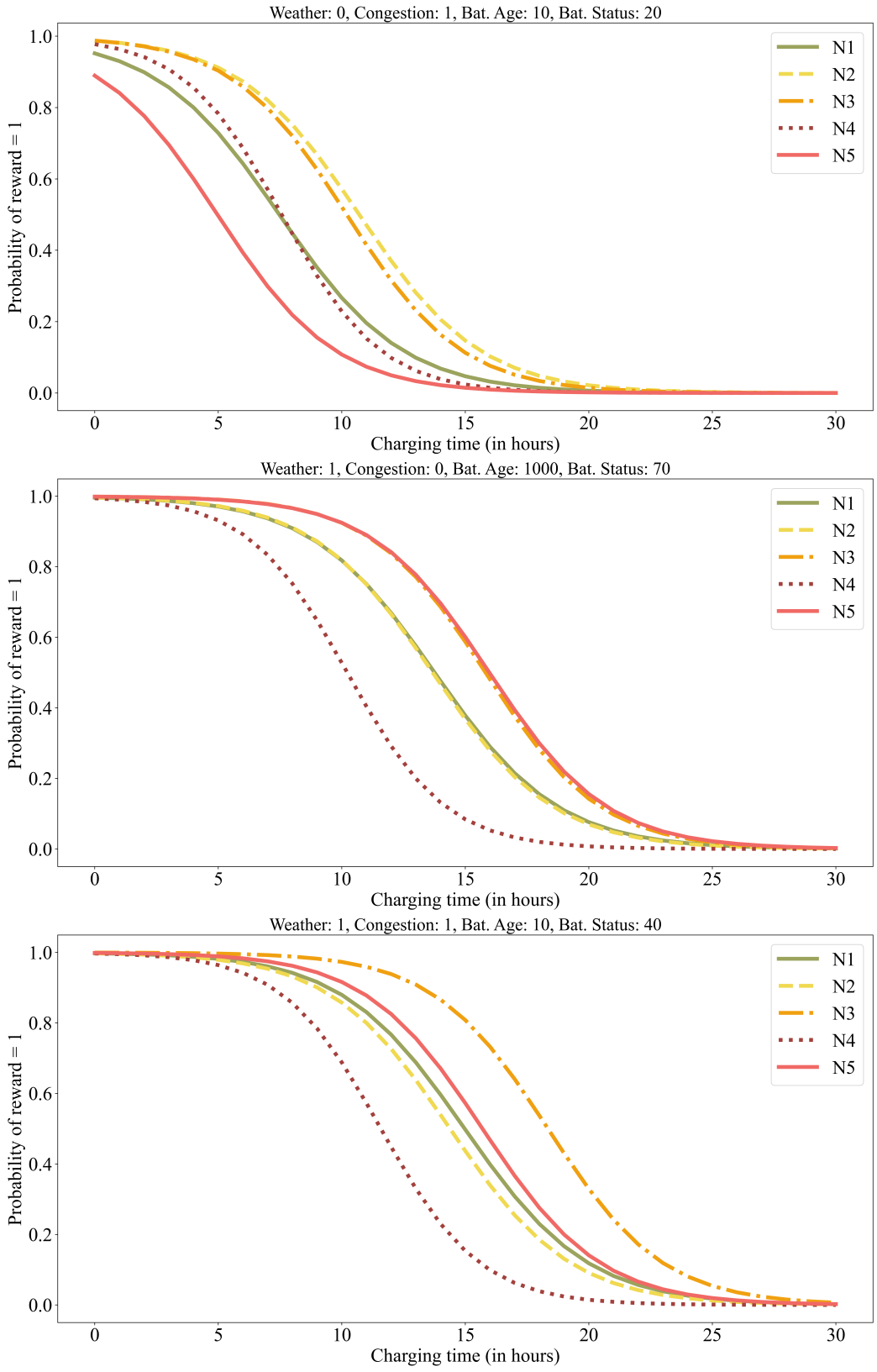

Figure 2. Real probabilities of obtaining a reward of 1 for each node and combination of factors.

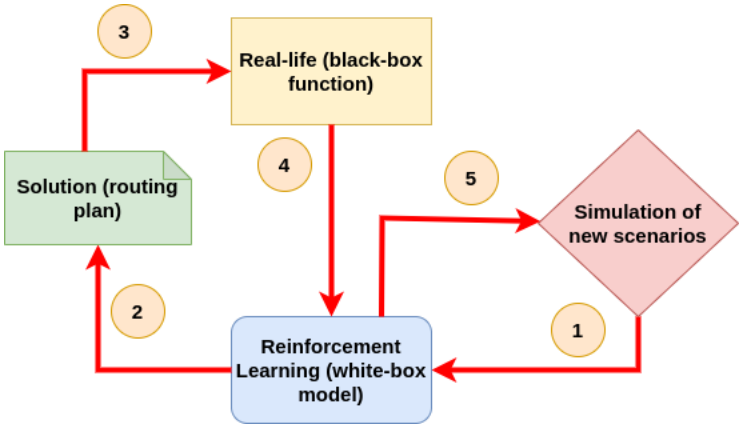

Figure 3. A conceptual schema of the proposed methodology.

6. Simulation and Algorithm Details

Using the Python programming language as a base, this section provides insights on the reinforcement learning algorithm employed to predict the probability of reward for each node and contextual conditions. It also describes the simulation process utilized to illustrate how the learning process works in a practical application. Notice that only the key parts of the code are provided here.

Listing 1 defines a multivariate logistic function, *real_reward_p*, which provides the real probability of obtaining a binary reward of 1 for a given node, based on the inputs weather conditions, congestion, battery age, expected battery, and charging time. This function acts as a black-box model (unknown for the learning algorithm) that emulates reality. This logistic function takes in a node, weather (0 or 1 where 1 is good weather and 0 is bad), congestion (0 or 1 where 1 is high congestion and 0 is low congestion), battery age (an integer between 10 and 1000), expected battery (an integer between 0 and 100), and charging time (an integer between 1 and 40) as parameters. Inside the function, a linear combination of the parameters and some weights stored in the *params* dictionary for that node are computed. The result of the linear combination is passed through the sigmoid function $1/(1 + np.exp(-linear\_comb))$ to obtain a probability of obtaining a reward of 1 for that node given the input parameters. For instance, if the parameters associated with a giving node i are $k_{1i} = 4$, $k_{2i} = -0.5$, $k_{3i} = -0.45$, $k_{4i} = -0.001$, $k_{5i} = -0.0004$, and $k_{6i} = 3.5$, then *real_reward_p(i, 1, 0, 20, 10, 20)* would return the probability of obtaining a reward of 1 for node i when the weather is good, congestion is low, the charging time is 40, the age of the battery is 10 cycles, and the battery status is 20.

In a similar fashion, Listing 2 defines a CatBoost model (gradient boosting with decision trees) [43] to predict the probabilities of obtaining a reward of 1 for a given node–region pair and context inputs. In order to train the model, the data are divided into two sets: training and testing. The training set is utilized to construct the model, while the test data serve the purpose of preventing overfitting by employing an early stopping strategy [44]. The model's performance improves significantly when more data are available. To ensure calibrated probabilities, Platt scaling is applied to fine-tune the model's probability outputs [45]. This model acts as a white-box one that aims at predicting how reality will behave, i.e., it tries to predict the real-life probability provided by the black-box function described in Listing 1. Given that the CatBoost Python library is employed, it is possible to use the inbuilt function *predict_proba* to generate predictions. Notice, however, that a different predictive model—e.g., a neural network—could have been used as a white box.

Listing 1. Python code illustrating the black box used to compute the real probability of reward.

```
# Logistic functions (one per node type) defining the probability of obtaining a binary reward of 1
# weather (1 = good, 0 = bad), congestion (1 = high, 0 = low), battery age (10 to 1000), time of charge (1 to 40), battery status (0 to 100)
# ex: p(reward = 1 / N1) = 1 / (1 + exp(-(4 * weather + -0.5 * congestion + -0.001 * battery age + -0.15 * expected battery + -0.0004 * time + 3.5)))
def real_reward_p(node, weather, congestion, batery_age, battery_status, time): % Attention AE: battery is spelt incorrectly here. The authors should check if this is correct.
    linear_comb = np.dot(params[node], [weather, congestion,batery_age, time, (20-abs(battery_status-20)), 1])
    return 1 / (1 + np.exp(-linear_comb)) # probability of obtaining a reward of 1
```

Listing 2. Python code illustrating the trained model used to predict the probability of reward.

```
# Train a CatBoost classifier to predict the probability of reward
# Split the data in train and test (80-20) and uses early stopping to avoid overfitting
# To predict probabilities correctly we apply Platt Scaling to calibrate the model
def fit_catboost_model(node, X, y):
    model = CatBoostClassifier(loss_function="Logloss",max_depth=5,iterations=1000, early_stopping_rounds=50, eval_metric="Logloss")
    node_X = X[X['node'] == node].drop(columns=['node'])
    node_y = y[X['node'] == node]
    node_y = [1 if i == True else 0 for i in node_y]
    X_train, X_val, y_train, y_val = train_test_split(node_X, node_y, test_size=0.2, random_state=42)
    model.fit(X_train, y_train, eval_set=(X_val, y_val), verbose=False)
    calibrated_model = CalibratedClassifierCV(model, cv='prefit', method='sigmoid')
    calibrated_model.fit(X_val, y_val)
    return calibrated_model
# Predict the probabilities of obtaining reward = 1 on each node for a new context
def predict_reward_p(node, weather, congestion, battery_age, battery_status, time):
    model = models[node]
    return model.predict_proba([[weather, congestion, battery_age, battery_status, time]])[0][1]
```

The code in Listing 3 defines a Python function called *select_node_eps* that selects a node from a dictionary *predicted_reward_p* based on a slightly modified epsilon-greedy strategy. The function takes in two optional parameters: *eps*, the value of epsilon used in the epsilon-greedy algorithm, and *uniform*, a Boolean flag that specifies whether to choose the node uniformly at random among all nodes with the same probability. The function first sorts the nodes in descending order based on their corresponding probability of reward. It then checks whether a random number between 0 and 1 is greater than *eps*. If it is, the function returns the node with the highest probability of reward (i.e., the first element in the sorted list). If the random number is less than or equal to *eps*, the function randomly selects a node from the sorted list, either uniformly or non-uniformly depending on the value of the uniform flag. If *uniform* is *True*, the function selects a node uniformly at random. If uniform is *False*, the function selects the node with the second highest probability of reward (i.e., the best 'alternative'). This function is typically used in reinforcement learning algorithms to balance the exploration and exploitation of different nodes based on their predicted rewards. Still, other strategies, such as the Thompson sampling [46], could also be used instead.

The code in Listing 4 simulates the generation of new regions in our route and selects the node to visit. The code uses a for loop to iterate over the number of observations specified by the variable *n_obs*. At each iteration, it simulates weather conditions, congestion, battery age, expected battery, and charging time. The code then selects nodes using a round-robin (balanced) selection method for the first *n_update* iterations. From that point on, it uses a trained model and the epsilon-greedy strategy to select the next node. The code then simulates the real probability of reward when this node is selected and updates the accumulated regrets. Since rewards are binary, the expected reward for each node is computed as the estimated probability of success. The code records the new data, increasing the size of observations for X (predictors), y (response), and z (regrets), and updates the models every *n_update* iterations.

Listing 3. Python code illustrating a modified epsilon-greedy strategy.

```
# Selects a node from nodes based on an epsilon-greedy strategy
def select_node_eps(predicted_reward_p, eps=0.1, uniform=True): # default epsilon
    # sort nodes from higher to lower probability of reward
    sorted_dict = dict(sorted(predicted_reward_p.items(), key=lambda x: x[1], reverse=
    True))
    sorted_list = list(sorted_dict.keys())
    if random.random() > eps:
        return sorted_list[0] # choose the node with the highest probability
    elif uniform:
        return random.choice(sorted_list)
    else: # choose the node with the second highest probability
        return sorted_list[1]
```

Listing 4. Python code illustrating the core part of the simulation.

```
# Simulate the generation of new routes and select the node
for i in range(n_obs):
    a_weather = random.randint([0, 1])
    a_congestion = random.randint([0, 1])
    a_battery_age = random.randint(10, 1000)
    a_battery_status = random.randint(0, 100)
    a_time = random.randint(start_time, end_time) # includes end_time
    # Round-robin (balanced) selection of first n_update nodes
    if i+1 < n_update:
        selected_node = list(nodes)[0]
        for node in nodes:
            if n_selected[node] < n_selected[selected_node]:
                selected_node = node
            # Caution!: for this stage, we'll use real p as predicted so we can
    estimate regrets
            predicted_reward_p[node] = real_reward_p(node, a_weather, a_congestion,
    a_battery_age, a_battery_status, a_time)
        n_selected[selected_node] = n_selected[selected_node] + 1
    else: # after the first model update, we can predict probabilities and select node
        for node in nodes:
            if i+1 == n_update: # set an initial model for node when n_update is
    reached
                models[node] = fit_catboost_model(node, X, y)
            predicted_reward_p[node] = predict_reward_p(node, a_weather, a_congestion,
    a_battery_age, a_battery_status, a_time)
        selected_node = select_node_eps(predicted_reward_p, 0.1, True)
    # Simulate the real probability of reward when this node is selected
    real_reward = random.random() < real_reward_p(selected_node, a_weather,
    a_congestion,a_battery_age, a_battery_status, a_time)
    # Update accum. regrets. Since rewards are binary, E[reward(node)] = estimated P(
    success / node)
    regrets = regrets + (max(predicted_reward_p.values()) - predicted_reward_p[
    selected_node])

    # Record the new data, thus increasing the size of observations for X, y, and z
    new_data = {'weather': a_weather, 'congestion': a_congestion,
                'battery_age':a_battery_age,
                'battery_status': a_battery_status,
                'time': a_time,'node': selected_node,
                'reward': real_reward,'regrets': regrets}

    new_df = pd.DataFrame(new_data, index=[0])
    df = pd.concat([df, new_df], ignore_index=True)
    X = df[['weather', 'congestion','battery_age','battery_status', 'time', 'node']]
    y = df['reward'].astype(bool) # make sure it is a bool data type

    # Update the models every n_update observations
    if (i+1) % n_update == 0 and i+1 > n_update:
        for node in nodes:
            models[node] = fit_catboost_model(node, X, y)
```

7. Computational Experiments

Table 2 shows a comparison between the predicted and the actual (real-life) probability of obtaining a reward of 1 for a randomly selected set of nodes and factor configurations. This table confirms that the trained model is capable of estimating the real-life probability with a relatively low average error.

After training the predictive model, it is possible to apply it to the proposed case study on dynamic OP with binary random rewards. Whenever the EV reaches a new region and receives updated contextual information such as weather, congestion, battery age, expected

battery, and time of charge, it will use the predictive model to select one of the nodes with a higher probability of obtaining a reward. Since selecting a node in one region might affect our options in subsequent regions, following a greedy approach in each region—i.e., always selecting the node with the highest expected reward at each step—would not necessarily lead us to an optimal solution for the entire trip. In a situation such as this, it is often convenient to employ diversification strategies based on biased randomization and agile optimization techniques [47].

Table 2. Predicted vs. actual probabilities of reward.

Node Type	Weather	Congest.	Batt. Age (Cycles)	Batt. Status (%)	Charge Time (h)	Predicted	Actual	Error
5	1	0	467	98	18	0.283	0.3974	0.1145
5	0	1	228	82	22	0.032	0.0007	0.0314
1	1	1	69	16	29	0.065	0.0034	0.0616
3	0	1	98	30	17	0.0499	0.0462	0.0038
1	0	0	107	79	25	0.0506	0.0014	0.0493
4	1	1	378	31	22	0.0273	0.0039	0.0234
5	0	1	486	3	15	0.0363	0.0101	0.0263
5	0	1	373	66	21	0.0309	0.0009	0.03
4	0	0	224	92	28	0.0262	0.0	0.0261
5	0	0	235	64	5	0.7532	0.6724	0.0808
1	1	1	354	22	23	0.0544	0.0278	0.0267
5	0	0	402	28	18	0.0312	0.0075	0.0237
3	0	1	196	31	3	0.9524	0.9465	0.0058
3	0	1	464	20	4	0.947	0.8929	0.054
3	1	0	58	44	4	0.9667	0.998	0.0313
4	0	0	234	59	29	0.0261	0.0	0.0261
4	1	1	314	59	7	0.7149	0.8833	0.1684
2	1	1	56	26	19	0.0691	0.1282	0.0591
5	1	1	159	52	28	0.0341	0.005	0.029
5	1	1	349	6	3	0.9603	0.9937	0.0334
Average:								0.0452

A total of 10,000 trips for the case study were simulated, which considered six regions and five types of nodes per region. Figure 4 shows boxplots with the expected accumulated reward for solutions generated by our approach (in which the trip is guided by the predictive model) and a non-guided approach (in which a node is randomly chosen in each region). Hence, while solutions proposed by our methodology show an average value of 2.83 and a standard deviation of 1.02 for the expected accumulated reward, solutions provided by a non-guided approach show an average value of 2.37 and a standard deviation of 0.99 for the expected accumulated reward. As expected, a *t*-test provides a *t*-statistic = 32.44, with an associated *p*-value = $4.82 * 10^{-225}$, which clearly indicates that the solutions generated by our reinforcement learning approach are significantly better than the ones constructed without taking into account the dynamic contextual conditions.

Figure 5 illustrates a comparative analysis between a non-guided solution and the route provided by our approach. The blue nodes (N1, N2, N3, N4, N5) in the figure represent the chargers in each region, and the weight of the edges in the graph represents

the real probabilities of reward based on the original solution. Notably, the selections made by our approach consistently demonstrate higher probabilities of obtaining rewards.

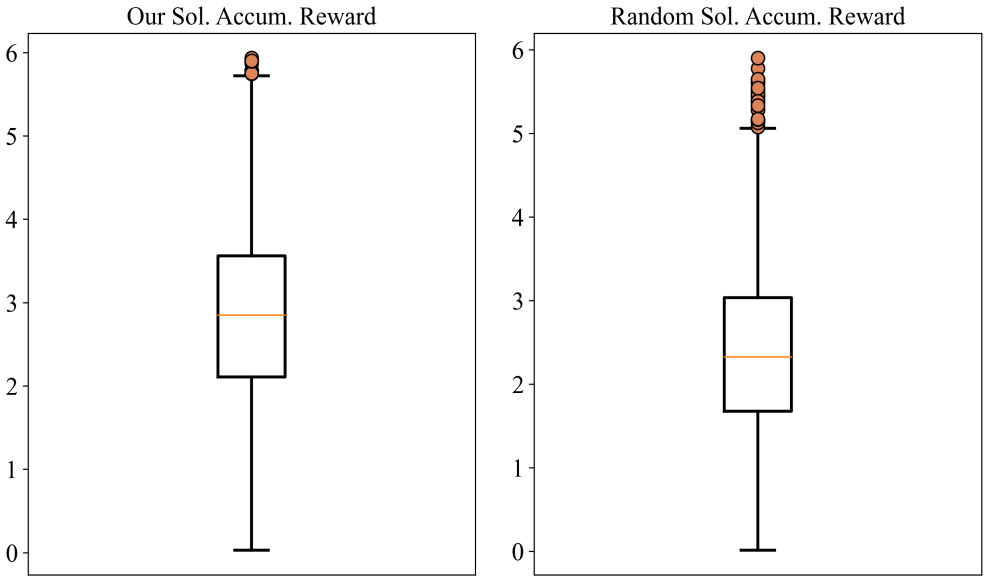

Figure 4. Boxplots for comparing our guided routing solutions with non-guided ones.

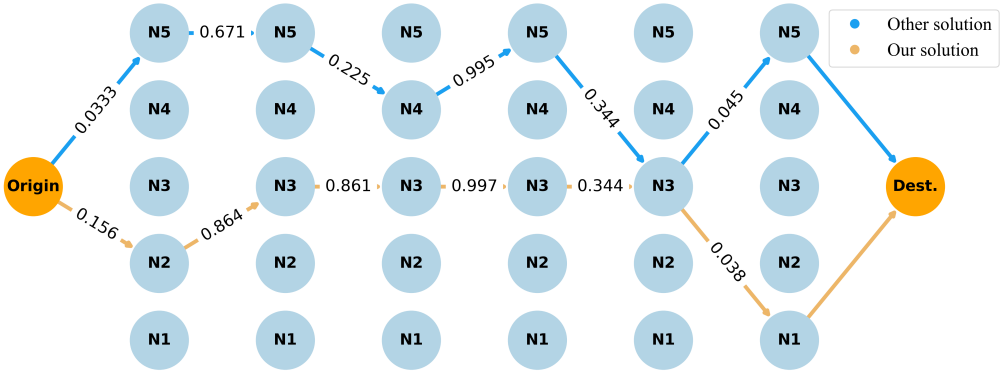

Figure 5. Visual representation of our guided routing solution and a non-guided one.

8. Conclusions

This paper proposes a hybrid methodology combining simulation and reinforcement learning to explore a vehicle orienteering problem with binary random rewards and dynamic conditions. This problem is discussed in the context of electric vehicles that, while covering a long trip, have to cross different regions and choose, in each of these region, a charging station with a high probability of reward. The challenge lies in the fact that the reward for visiting each charging node is a binary random variable and depends on dynamic context conditions, such as: weather conditions, road or on-site congestion, current battery status, etc. In order to emulate the learning process in real-life conditions under uncertainty, a simulation is employed. Thus, a black-box model is used to estimate the real-life probability of obtaining a reward for each node based on the dynamic context conditions at a given time. A reinforcement learning mechanism is then employed to make informed decisions at each stage of the problem, and a logistic regression model is used to predict the aforementioned probabilities. Through the simulation results, it is shown that the proposed reinforcement learning approach can effectively learn to make informed

decisions based on the dynamic context conditions. The computational experiments show that a statistically significant improvement is obtained when the proposed approach is utilized. All in all, the proposed approach can be useful in a range of real-life scenarios such as transportation logistics, delivery services, and resource allocation.

One potential direction is to explore more complex models for the probability of obtaining a reward, such as neural networks or models that account for additional context conditions. To mitigate some of the downsides of the strategy chosen in this paper (ε greedy), various enhancements could be carried out, such as using decaying exploration rates or dynamically adjusting the exploration rate based on the agent's learning progress. Another potential area for future work is to investigate the impact of different reinforcement learning algorithms and strategies on the performance of the model. For example, analyzing different exploration strategies (beyond the epsilon-greedy strategy used in this study) may lead to more efficient learning and better decision making. Furthermore, it may be valuable to consider the use of more advanced optimization techniques, such as metaheuristics and simheuristics [48], that can be combined with the reinforcement learning approach introduced in this paper. Finally, this study focused on a single vehicle traveling through a fixed set of regions. Future work could explore more complex scenarios, such as the involvement of multiple vehicles, i.e., a dynamic team orienteering problem with stochastic rewards.

Author Contributions: Conceptualization, A.A.J.; methodology, A.A.J., Y.A., and J.P.; software, Y.A., C.A.M. and X.A.M.; validation, J.P. and X.A.M.; writing—original draft preparation, A.A.J., C.A.M. and R.F.; writing—review and editing, C.A.M., Y.A. and R.F.; supervision, J.P., X.A.M. and A.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the European Commission projects SUN (HORIZON-CL4-2022-HUMAN-01-14-101092612), and AIDEAS (HORIZON-CL4-2021-TWIN-TRANSITION-01-07-101057294).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EV	Electric vehicle
BMS	Battery management system
OP	Orienteering problem
AC	Alternating current
DC	Direct current
RL	Reinforcement learning
HEV	Hybrid electric vehicle
BEV	Battery electric vehicle
SOC	State of charge
V2G	Vehicle-to-grid

References

1. Almouhanna, A.; Quintero-Araujo, C.L.; Panadero, J.; Juan, A.A.; Khosravi, B.; Ouelhadj, D. The location routing problem using electric vehicles with constrained distance. *Comput. Oper. Res.* **2020**, *115*, 104864. [CrossRef]
2. Abdollahi, A.; Han, X.; Avvari, G.V.; Raghunathan, N.; Balasingam, B.; Pattipati, K.K.; Bar-Shalom, Y. Optimal battery charging, Part I: Minimizing time-to-charge, energy loss, and temperature rise for OCV-resistance battery model. *J. Power Sources* **2015**, *303*, 388–398. [CrossRef]
3. Golden, B.; Levy, L.; Vohra, R. The Orienteering Problem. *Nav. Res. Logist.* **1987**, *34*, 307–318. [CrossRef]

4. Panadero, J.; Currie, C.; Juan, A.A.; Bayliss, C. Maximizing Reward from a Team of Surveillance Drones under Uncertainty Conditions: A Simheuristic Approach. *Eur. J. Ind. Eng.* **2020**, *14*, 1–23. [CrossRef]
5. Bayliss, C.; Juan, A.A.; Currie, C.S.; Panadero, J. A learnheuristic approach for the team orienteering problem with aerial drone motion constraints. *Appl. Soft Comput.* **2020**, *92*, 106280. [CrossRef]
6. Bilgin, E. *Mastering Reinforcement Learning with Python: Build Next-Generation, Self-Learning Models using Reinforcement Learning Techniques and Best Practices*; Packt Publishing Ltd.: Birmingham, UK, 2020.
7. IPCC. *Climate Change 2022: Mitigation of Climate Change*; Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change; IPCC: Geneva, Switzerland, 2022.
8. IEA. *Global EV Outlook 2023*; International Energy Agency: Paris, France, 2023.
9. Dixit, M.; Muralidharan, N.; Parejiya, A.; Essehli, R.; Belharouak, I.; Amin, R. Electrochemical energy storage systems. In *Emerging Trends in Energy Storage Systems and Industrial Applications*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 259–282.
10. Liang, J.; Wu, T.; Wang, Z.; Yu, Y.; Hu, L.; Li, H.; Zhang, X.; Zhu, X.; Zhao, Y. Accelerating perovskite materials discovery and correlated energy applications through artificial intelligence. *Energy Mater* **2022**, *2*, 200016. [CrossRef]
11. Qi, X.; Wu, G.; Boriboonsomsin, K.; Barth, M.J. Data-driven decomposition analysis and estimation of link-level electric vehicle energy consumption under real-world traffic conditions. *Transp. Res. Part D Transp. Environ.* **2018**, *64*, 36–52. [CrossRef]
12. Kwang, H.N. *AC Motor Control and Electric Vehicle Applications*; CRC Press, Taylor & Francis Group: Boca Raton, FL, USA, 2010.
13. Alanazi, F. Electric Vehicles: Benefits, Challenges, and Potential Solutions for Widespread Adaptation. *Appl. Sci.* **2023**, *13*, 6016. [CrossRef]
14. Böhme, T.J.; Frank, B. *Hybrid Systems, Optimal Control and Hybrid Vehicles. Theory, Methods and Applications*; Springer International Publishing: Cham, Switzerland, 2017.
15. Townsend, A.; Gouws, R. A Comparative Review of Lead-Acid, Lithium-Ion and Ultra-Capacitor Technologies and Their Degradation Mechanisms. *Energies* **2022**, *15*, 4930. [CrossRef]
16. Chau, K. Pure electric vehicles. In *Alternative Fuels and Advanced Vehicle Technologies for Improved Environmental Performance*; Woodhead Publishing: Cambridge, UK, 2014; Volume 8176, pp. 655–684.
17. Guo, F.; Wu, X.; Liu, L.; Ye, J.; Wang, T.; Fu, L.; Wu, Y. Prediction of remaining useful life and state of health of lithium batteries based on time series feature and Savitzky-Golay filter combined with gated recurrent unit neural network. *Energy* **2023**, *270*, 126880. [CrossRef]
18. Li, W.; Stanula, P.; Egede, P.; Kara, S.; Herrmann, C. Determining the main factors influencing the energy consumption of electric vehicles in the usage phase. *Procedia CIRP* **2016**, *48*, 352–357. [CrossRef]
19. Bi, J.; Wang, Y.; Zhang, J. A data-based model for driving distance estimation of battery electric logistics vehicles. *EURASIP J. Wirel. Commun. Netw.* **2018**, *251*, 1–13. [CrossRef]
20. Preger, Y.; Barkholtz, H.M.; Fresquez, A.; Campbell, D.L.; Juba, B.W.; Romàn-Kustas, J.; Ferreira, S.R.; Chalamala, B. Degradation of Commercial Lithium-Ion Cells as a Function of Chemistry and Cycling Conditions. *J. Electrochem. Soc.* **2020**, *167*, 1–9.
21. Koltermann, L.; Cortés, M.C.; Figgener, J.; Zurmühlen, S.; Sauer, D.U. Power Curves of Megawatt-Scale Battery Storage Technologies for Frequency Regulation and Energy Trading. *Appl. Energy* **2023**, *347*, 121428. [CrossRef]
22. Jiang, J.; Shi, W.; Zheng, J.; Zuo, P.; Xiao, J.; Chen, X.; Xu, W.; Zhang, J.G. Optimized Operating Range for Large-Format LiFePO4/Graphite Batteries. *J. Electrochem. Soc.* **2013**, *161*, 336–341. [CrossRef]
23. Ameli, M.T.; Ameli, A. Electric vehicles as means of energy storage: Participation in ancillary services markets. In *Energy Storage in Energy Markets. Uncertainties, Modelling, Analysis and Optimization*; Academic Press: Cambridge, MA, USA, 2021; pp. 235–249.
24. Mastoi, M.S.; Zhuang, S.; Hafiz Mudassir, M.; Haris, M.; Hassan, M.; Usman, M.; Bukhari, S.S.H.; Ro, J.S. An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends. *Energy Rep.* **2022**, *8*, 11504–11529. [CrossRef]
25. Yong, J.Y.; Ramachandaramurthy, V.K.; Tan, K.M.; Mithulananthan, N. A Review on the State-of-the-Art Technologies of Electric Vehicle, Its Impacts and Prospects. *Renew. Sustain. Energy Rev.* **2015**, *49*, 365–385. [CrossRef]
26. U.S. Department of Transportation. Charging Speeds. Rural Electric Vehicle Toolkit. Available online: https://www.transportation.gov/rural/ev/toolkit/ev-basics/charging-speeds (accessed on 1 August 2023).
27. Brenna, M.; Foiadelli, F.; Leone, C.; Longo, M. Electric Vehicles Charging Technology Review and Optimal Size Estimation. *J. Electr. Eng. Technol.* **2020**, *15*, 2539–2552. [CrossRef]
28. Solntsev, A.; Asoyan, A.; Nikitin, D.; Bagrin, V.; Fediushkina, O.; Evtykov, S.; Marusin, A. Influence of temperature on the performance and life cycle of storage batteries. *Transp. Res. Procedia* **2021**, *57*, 652–659. [CrossRef]
29. Zhang, Z.; Li, W.; Zhang, C.; Chen, J. Climate control loads prediction of electric vehicles. *Appl. Therm. Eng.* **2017**, *110*, 1183–1188. [CrossRef]
30. Yang, X.G.; Zhang, G.; Ge, S.; Wang, C.Y. Fast charging of lithium-ion batteries at all temperatures. *Appl. Therm. Eng.* **2018**, *115*, 7266–7271. [CrossRef] [PubMed]
31. Jonas, T.; Hunter, C.D.; Macht, G.A. Quantifying the Impact of Traffic on Electric Vehicle Efficiency. *World Electr. Veh. J.* **2022**, *13*, 15. [CrossRef]
32. Hu, K.; Wu, J.; Schwanen, T. Differences in Energy Consumption in Electric Vehicles: An Exploratory Real-World Study in Beijing. *J. Adv. Transp.* **2017**, *2017*, 1–17.
33. Ilhan, T.; Iravani, S.; Daskin, M. The Orienteering Problem with Stochastic Profits. *IIE Trans.* **2008**, *40*, 406–421. [CrossRef]

34. Campbell, A.; Gendreau, M.; Thomas, B. The Orienteering Problem with Stochastic Travel and Service Times. *Ann. Oper. Res.* **2011**, *186*, 61–81. [CrossRef]
35. Papapanagiotou, V.; Montemanni, R.; Gambardella, L. Objective Function Evaluation Methods for the Orienteering Problem with Stochastic Travel and Service Times. *J. Appl. Oper. Res.* **2014**, *6*, 16–29.
36. Verbeeck, C.; Vansteenwegen, P.; Aghezzaf, E.H. Solving the Stochastic Time-Dependent Orienteering Problem with Time Windows. *Eur. J. Oper. Res.* **2016**, *255*, 699–718.
37. Evers, L.; Glorie, K.; van der Ster, S.; Barros, A.; Monsuur, H. A Two-Stage Approach to the Orienteering Problem with Stochastic Weights. *Comput. Oper. Res.* **2014**, *43*, 248–260. [CrossRef]
38. Lau, H.C.; Yeoh, W.; Varakantham, P.; Nguyen, D.T.; Chen, H. Dynamic Stochastic Orienteering Problems for Risk-Aware Applications. *arXiv* **2012**, arXiv:1210.4874.
39. Varakantham, P.; Kumar, A. Optimization Approaches for Solving Chance-Constrained Stochastic Orienteering Problems. In *Algorithmic Decision Theory*; Springer: Cham, Switzerland, 2013; Volume 8176, pp. 387–398.
40. Zhang, S.; Ohlmann, J.; Thomas, B. A Priori Orienteering with Time Windows and Stochastic Wait Times at Customers. *Eur. J. Oper. Res.* **2014**, *239*, 70–79. [CrossRef]
41. Gama, R.; Fernandes, H.L. A reinforcement learning approach to the orienteering problem with time windows. *Comput. Oper. Res.* **2021**, *133*, 105357. [CrossRef]
42. Gunawan, A.; Lau, H.; Vansteenwegen, P. Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications. *Eur. J. Oper. Res.* **2016**, *255*, 315–332. [CrossRef]
43. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
44. Yao, Y.; Rosasco, L.; Caponnetto, A. On early stopping in gradient descent learning. *Constr. Approx.* **2007**, *26*, 289–315. [CrossRef]
45. Platt, J. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.* **2000**, *10*.
46. Umami, I.; Rahmawati, L. Comparing Epsilon Greedy and Thompson Sampling Model for Multi-Armed Bandit Algorithm on Marketing Dataset. *J. Appl. Data Sci.* **2021**, *2*, 14–26. [CrossRef]
47. Do C. Martins, L.; Hirsch, P.; Juan, A.A. Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *Int. Trans. Oper. Res.* **2021**, *28*, 201–221.
48. Chica, M.; Juan, A.A.; Bayliss, C.; Cordón, O.; Kelton, W.D. Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation. *SORT-Stat. Oper. Res. Trans.* **2020**, *44*, 311–334. [CrossRef]