Intelligent Trajectory Design and Charging Scheduling in Wireless Rechargeable Sensor Networks with Obstacles

Journal:	*Transactions on Mobile Computing*
Manuscript ID	TMC-2023-04-0377
Manuscript Type:	Regular
Keywords:	Wireless rechargeable sensor networks, Mobile charging, Trajectory design, Reinforcement learning

SCHOLARONE™
Manuscripts

Intelligent Trajectory Design and Charging Scheduling in Wireless Rechargeable Sensor Networks with Obstacles

Xiuling Zhang, Riheng Jia, *Member, IEEE, ACM,* Quanjun Yin, Zhonglong Zheng, *Member, IEEE,* and Minglu Li, *Fellow, IEEE*

Abstract—Wireless rechargeable sensor networks (WRSNs) are promising in maintaining sustainable large-area monitoring tasks. Mobile chargers (MCs) are commonly used in WRSNs to replenish energy to nodes due to its flexibility and easy maintenance. Most existing works on WRSNs focus on designing offline or model-based online charging methods, which need the exact system information to conduct the optimization. However, in practical WRSNs, the exact system information such as the nodes' locations and energy consumption rates may not be easily accessible to the optimizer due to their unpredictability and high dynamics. Thus, in this work, we jointly optimize the MC's trajectory design and charging scheduling in a general and practical WRSN with inaccessibility to the exact system information, such that the charging utility of the MC is maximized. To address this problem, we introduce the model-free reinforcement leanring (RL) technique, which enables the MC to learn to jointly optimize its moving trajectory and charging scheduling by interacting with the environment and tracking feedback signals from nodes and obstacles in real time. Specifically, we develop a soft actor-critic based mobile security policy intervened algorithm (SAC-MSPI) based on a novel safe RL framework, which maximizes the MC's charging utility while maintaining the safe movement (not hitting obstacles) for the MC during the entire charging period. Extensive evaluation reuslts show that the proposed SAC-MSPI algorithm outperforms existing main RL solutions and traditional baseline algorithms with respect to the charging utility maximization as well as the collision avoidance.

Index Terms—Wireless rechargeable sensor networks, trajectory design, charging scheduling, reinforcement learning.

1 INTRODUTION

1.1 Background

Since the pioneering work on the resonant coupling based wireless power transfer (WPT) [1], there have been increasing researches on investigating how the WPT technique helps address the energy bottleneck in wireless sensor networks (WSNs) [2], [3], [4]. Because WSNs usually consist of sensors with the limited battery capacity and it is costly to manually change their batteries from time to time, especially when the WSN is deployed in some hazardous areas. Although many literatures focus on applying the energy harvesting (EH) technique to WSNs to make sensors harvest energy from environmental energy sources [5] [6], the problem of low and fluctuated energy harvesting rate remains unsolved. To replenish energy to sensor nodes (nodes) in WSNs by the WPT technique, we either deploy static chargers within the network or dispatch mobile chargers (MC-s) to periodically visit nodes with rechargeable batteries, which may prolong the network life time or even maintain sustainable network operations [7]. MCs are flexible, cost-efficient and easy-maintainable, compared with static chargers, which thus gains the growing popularity in academia in recent years [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31]. Since the MC carries the limited amount of onboard energy and can only successfully charge nodes within its finite charging range, the inappropriate moving trajectory of the MC may cause the low charging utility, the long charging delay, the high energy cost and so on. In addition, the MC's moving trajectory must be carefully designed to avoid colliding with obstacles considering the complex environment where WSNs are usually deployed [23], [32]. Thus, jointly optimizing the MC's moving trajectory and charging scheduling is a primary issue for maintaining the normal operation of nodes in WSNs.

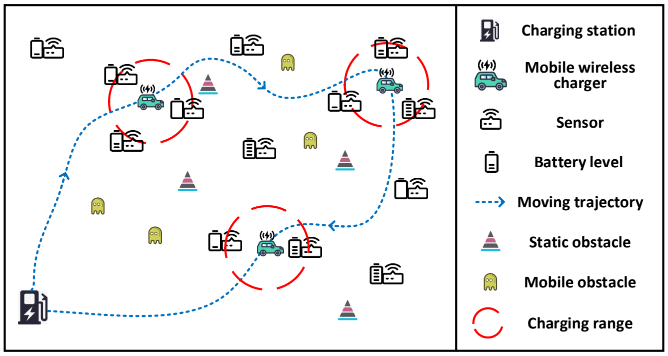


Fig. 1. Network Model.

1.2 Snapshot of Our Work and Associated Challenges

In this work, we investigate the charging utility maximization problem in a wireless rechargeable sensor network

- X. Zhang and Q. Yin are with the College of System Engineering, National University of Defense Technology, Changsha, China.
- R. Jia, Z. Zheng and M. Li are with the School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China.
- The corresponding author is Riheng Jia, Email: rihengjia@zjnu.edu.cn.

(WRSN), by jointly optimizing the MC's moving trajectory and charging scheduling. The network model is illustrated in Fig. 1. Nodes are randomly distributed within the network and each node may randomly consume a certain amount of energy in each time slot due to the node operation (e.g., sensing and communication). Thus the real-time residual battery level of each node may vary over slots. To replenish energy to nodes for maintaining their normal operation, a MC is dispatched to periodically visit these nodes. For a particular charging tour, the MC starts from the charging station, then moves within the network and finally returns to the charging station. During the charging tour, the MC can stop anywhere and anytime to charge nodes. At each stop point (charging location), the MC spends some time charging the nodes located within its charging range by the omnidirectional WPT technique, i.e., all the nodes located within the MC's charging range can be charged simultaneously. The detailed charging model is defined in (4). In addition, we assume that there are both static and mobile obstacles within the network and thus the MC needs to avoid colliding with obstacles while moving within the network. Otherwise, the MC may be damaged by hitting obstacles, which fails the charging task. Our goal is to maximize the total amount of effective charged energy during a charging tour, via jointly optimizing the MC's moving trajectory and charging scheduling. Note that the effective charged energy refers to the energy which is really received by the node through the MC-to-node wireless charging, excluding the energy wasted due to the node's battery overflow. *We identify the main challenges of solving this problem as follows.*

The exact node-side information is inaccessible to the MC: In this work, we assume that the exact node-side information, i.e., the exact location and energy consumption rate of each node, is inaccessible to the MC. In practical scenarios, it is costly to equip each node with the global positioning system (GPS) module. Also the GPS signal may usually be blocked in complex environments. The corresponding energy consumption for maintaining the GPS connection cannot be ignored, considering the small battery capacity of nodes. Since the workload of each node may vary over time (depends on the stochastic event in the environment), which makes the energy consumption rate of each node fluctuate randomly over time. Thus the MC cannot know how the energy consumption rate of each node changes over time. Without the exact node-side information, traditional optimization methods can not solve (or even explicitly formulate) our problem.

The tradeoff between the charging utility and the movement safety: To maximize the charging utility, i.e., the total amount of effective charged energy, the MC needs to move to some targeted areas to charge nodes by comprehensively considering nodes' initial battery levels, energy consumption rates and distributed density within the network. At the same time, the MC needs to avoid colliding with obstacles while moving within the network. It is risky for the MC to move to the area with the high achievable charging utility as well as the high probability of hitting obstacles. Thus it is challenging to maximize the charging utility and reduce the risk of hitting obstacles at the same time, especially when the MC cannot know the exact location of each obstacle.

The infinite searching space of MC's moving trajectory: Since the MC can move freely and stop anywhere to charge nodes within the whole network area, which results in the infinite searching space of the MC's moving trajectory in a continuous two-dimensional (2D) plane.

To tackle the above challenges, we introduce the reinforcement learning (RL) technique, which enables the MC to learn to jointly optimize its moving trajectory and charging scheduling in a trial-and-error way, by interacting with the environment and tracking feedback signals from nodes and obstacles in real time during the charging tour. Specifically, we model the joint trajectory design and charging scheduling problem as a Markov decision process (MDP), where the MC optimizes a series of sequential decisions (i.e., the movement decision and the charging decision in each slot) to maximize the total rewards (charging utility) obtained during the charging tour. Based on the formulated MDP, we develop the soft actor-critic based mobile security policy intervened algorithm (SAC-MSPI) based on a novel safe RL framework [33]. The proposed SAC-MSPI algorithm decouples the charging utility maximization and the collision risk minimization, which improves existing RL solutions by not sacrificing the charging utility for the safety of movement. We refer the reader to Section 5 for the detail of SAC-MSPI. In addition, the SAC module can well handle the continuous action (i.e., move and charge) space by outputting a certain probability distribution of actions instead of a single action. *The main contributions of this work are summarized as follows.*

- To our best knowledge, this is the first attempt to jointly optimize the MC's moving trajectory and charging scheduling in a general WRSN without accessibility to the exact system information such as the node's location and energy consumption rate and the obstacle's location.
- We develop the SAC-MSPI algorithm based on a novel safe RL framework, which maximizes the average effective charging rate of the MC while maintaining the safe movement during the charging tour. Extensive evaluations validate the superiority of our designed algorithm, compared with existing main RL algorithms and traditional baseline algorithms.
- We evaluate the performance of SAC-MSPI under various experiment scenarios. Results show that the MC can always find a safe moving trajectory with the maximum achievable charging utility by using the designed SAC-MSPI algorithm, which proves its stability.

The remainder of this work is organized as follows. Section 2 introduces the related works. Section 3 and Section 4 illustrate the network model and problem formulation respectively. Section 5 presents the architecture and implementation of SAC-MSPI. Section 6 evaluates the performance of SAC-MSPI. Section 7 concludes this work.

2 Related Works

In this section, we compare our work with existing related literatures on WRSNs as follows, which helps clarify the novelty of our work better.

2.1 Non-Machine Learning based Optimization in WRSNs

In this part, we mainly introduce the research works on modeling and performance optimization in WRSNs when the whole system information is fully accessible by the scheduler.

The deployment optimization of static chargers: To replenish energy to nodes with static chargers, it is essential to decide the exact fixed location of each charger within the network. For example, He et al. [34] studied how to appropriately deploy a number of static wireless chargers within a 2D plane to guarantee that the aggregated energy receiving rate at each node is not less than the corresponding energy consumption rate. Zhang et al. [35] jointly optimized the charger placement and transmission power control to maximize the overall charging utility of all nodes within the network. Further, Wang et al. [36] and Dai et al. [37], [38], [39], [40] extended the problem of static charger placement to the case of directional WPT. In addition, Dai et al. [41], [42] investigated the safe charging problem of jointly optimizing the charger placement and power control so that no location in the network has electromagnetic radiation exceeding a given threshold while guaranteeing the charging utility. In this work, we focus on the trajectory design and charging scheduling of a mobile charger, which is different from those works on static chargers placement.

The trajectory optimization of mobile chargers: There have been a significant amount of literatures on studying the trajectory design of mobile chargers in WRSNs with various objectives. Some of them [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] focused on maximizing the MC's charging utility in WRSNs. Specifically, they usually assumed that the charging utility for charging each node depends on the actual amount of energy received by the node, which was commonly formulated as a specific utility function. These charging utility maximization problems were usually reduced into submodular maximization problems with different network constraints. Then some greedy-based algorithms were designed to solve these submodular maximization problems with provable approximation ratios. Some researchers focused on investigating the problem of minimizing the charging delay in WRSNs [24], [25], [26], [27]. In particular, the works [24], [25], [26] minimized the total charging time used until when the energy demand of each node is satisfied, without considering the travelling time of the MC. Xu et al. [27] extended the problem of charging delay minimization to the case of multiple mobile chargers, where the target is to find a closed charging tour for each of the multiple deployed chargers, such that the longest charging delay of all the charging tours is minimized. Wang et al. [31] minimized the energy cost of the MC by jointly optimizing the charging delay and trajectory length. Jia et al. [43] extended the problem of MC's energy cost minimization to the general case when the node distribution as well as the distribution of charging demands among nodes is arbitrary. In general, most of above works on mobile charging in WRSNs conducted the modeling and optimization using offline scheduling methods which assign the deterministic charging task to the MC in advance. Some online charging methods [19], [30], [44], [45] focused on the on-demand charging paradigm and design the MC's moving trajectory based on the real-time information of charging environments. Both of those offline and online charging methods assumed that the whole system information is fully accessible to the MC, i.e., the MC exactly knows how the system changes in the future. However, in practice, the exact system information such as nodes' locations and energy consumption rates and obstacles' locations might not be easily accessible (or even reliably predicted through a explicit model) due to their dynamics.

2.2 Machine Learning based Optimization in WRSNs

A few previous works investigated how to incorporate RL techniques to design the MC's moving trajectory in WRSNs. For example, Cao et al. [46] designed the RL-based charging scheduling algorithm to minimize the trajectory length of the MC and the number of dead nodes in on-demand charging paradigm. Chen et al. [47] studied the joint sensor activation and mobile charging vehicle scheduling in a WRSN-based industrial internet of things, which was solved by developing a novel scheme integrating RL and marginal product based approximation algorithms. We would like to point out that both of works [46] and [47] assumed that the knowledge of nodes' exact locations and energy consumption rates is known in advance. Liu et al. [48] studied the charging reward maximization problem with the non-deterministic mobility model, where each mobile device visits some hotspots probabilistically. A novel model-free RL approach was designed to determine the MC's next hotspot to visit and the corresponding staying time, given the current system environment. Liang et al. [49] studied the cooperative charging problem in WRSNs with multiple mobile chargers and proposed an RL-based algorithm called asynchronous and scalable multi-agent proximal policy optimization to jointly maximize the charging utility and minimize the number of dead nodes. Compared with the works [48] and [49], in this work, we consider a more general and practical system model specified as follows: 1) There are both static and mobile obstacles in the WRSN; 2) The energy consumption rate of each node presents spatial-temporal dynamics within the network and the energy transfer process is affected by both the charging distance and the charging time; 3) The exact location information of nodes and obstacles is unknown to the MC before the charging task. The general and practical model makes the problem-solving more complicated and introduces new challenges such as handling the tradeoff between maximizing the charging utility and reducing the risk of hitting obstacles.

3 Network Model

We define the WRSN as a 2D rectangular area where a number of sensors nodes (nodes) $\mathcal{S} = \{s_1, s_2, ..., s_N\}$ are randomly distributed, where N is the total number of nodes. We also assume that some static obstacles are randomly distributed within this area and some mobile obstacles may move randomly within this area. Each node s_i is powered by a rechargeable battery with the same maximum capacity B_s. We assume that the time is divided into equal-length slots. At each slot t, each node s_i may consume a certain amount of energy e_i^t, which is randomly, identically and independently sampled from a certain probability distribution $\mathcal{E}_i$, where $i = 1, 2, ..., N$. A mobile charger (MC) is dispatched to periodically visit these nodes to recharge their batteries, for maintaining the normal operation of the WRSN. For example in Fig. 1, the MC starts from the charging station, moves within the network while providing wireless charging services to nodes and finally returns to the charging station to recharge itself. We assume that the MC is equipped with a main battery and a backup battery. When the main battery is drained out during the charging journey, the MC uses the backup battery to return to the charging station directly. We assume that the energy stored in the backup battery is sufficient for the MC to return to the charging station from anywhere within the network[1]. In the following, we first introduce two network models and then present the problem formulation. We summarize the main system parameters in Table 1.

TABLE 1
MAIN SYSTEM PARAMETERS

Symbols	Definitions
$\mathcal{S}$	The set of nodes in the network
s_i^t	The i-th node charged at slot t
v_{max}	The maximum speed component of the MC
B_m	The main battery capacity of the MC
B_s	The battery capacity of each node
$\mathcal{P}$	A particular moving trajectory
$T_{\mathcal{P}}$	The total time consumed on $\mathcal{P}$
$\Lambda_{\mathcal{P}}$	The set of nodes charged on $\mathcal{P}$
v_t	The moving speed of the MC in slot t
o_t	The moving direction of the MC in slot t
v_t^x	The X-axis speed of the MC in slot t
v_t^y	The Y-axis speed of the MC in slot t
c_t	The MC's moving cost in slot t
δ	The MC's moving cost per unit distance
l_t	The MC's location at the beginning of slot t
R	The charging range of the MC
E_t	the total effective charged energy in slot t
p_c^t	The allocated charging power in slot t
$p_r^t(i)$	The receiving power at the node s_i^t
$D_{\mathcal{P}}$	The total length of $\mathcal{P}$

3.1 Mobility Model and Trajectory

We assume that the MC can move freely within the 2D rectangular area. Note that since there are both static and mobile obstacles in the 2D area, the MC needs to be careful to avoid colliding with obstacles, which we will analyze in detail later. At the beginning of each slot t, the MC decides the moving direction o_t as well as the moving speed v_t within slot t according to a certain control policy π, where $o_t \in [0, 360°]$. We assume that v_t is divided into two components, i.e., the X-axis speed v_t^x and the Y-axis speed v_t^y, where we have

$$v_t = \sqrt{\left(v_t^x\right)^2 + \left(v_t^y\right)^2}, v_t^x, v_t^y \in \left[-v_{max}, v_{max}\right]. \tag{1}$$

Thus in fact the moving direction o_t is inherently decided by the two speed components v_t^x and v_t^y. For simplicity, we assume that the length of each slot is one unit time and thus the moving distance per slot is v_t. We denote δ as the MC's moving cost (energy used for movement) per unit distance and the moving cost of the MC in slot t is $c_t = \delta v_t$. Note that the moving cost in slot t cannot exceed the MC's residual battery level b_m^t at the beginning of slot t, i.e., we have $c_t \leq b_m^t$. We denote the location of the MC at the beginning of slot t as a 2D coordinate $l_t = [x_t, y_t]$ and the real-time location of the MC is updated as

$$\begin{aligned} x_{t+1} &= x_t + v_t^x, \\ y_{t+1} &= y_t + v_t^y. \end{aligned} \tag{2}$$

We define $\mathcal{P}$ as the MC's moving trajectory which starts at the charging station and ends at the location when the main battery of the MC is just drained out. We also define $T_{\mathcal{P}}$ as the total time (total number of slots) consumed on $\mathcal{P}$. In this work, our goal is to find the optimal moving trajectory $\mathcal{P}^*$ as well as the optimal charging scheduling to maximize the average effective charging rate, which is defined in (8). In particular, we assume that the exact location information of each node is not accessible by the MC before the charging tour. Also, the MC cannot know the location of each static obstacle and how the mobile obstacle moves within the network before the charging tour. These assumptions are made to accommodate to practical scenarios when nodes are deployed in complex, unstructured and dynamical environments such as battle fields and disaster areas. Under these assumptions, the MC needs to learn to optimize $\mathcal{P}$ by interacting with the environment during the charging tour.

Since both static and mobile obstacles exist in the 2-D area, the MC should be careful while moving to avoid colliding with the obstacles. Otherwise, the MC may be easily damaged, which results in the failure of the charging task. We assume that the size of each obstacle is much larger than that of the MC, since it may not cause the serious damage even when the MC hits the small obstacle (e.g., small rocks and little animals). Based on this assumption, the size of the MC can be ignored and we only consider the size of the obstacle. Specifically, we denote o_j^t as the location of the center point of the obstacle O_j in slot t within the 2-D area. We also denote r_o as the maximum distance between the center point and any point on the surface of each obstacle. We assume that the MC can detect any obstacle within a certain range r_d by using the ranging radar. In each slot t, if the distance between the MC and the center point of an obstacle is smaller than or equal to r_o, i.e., $\left|l_t - o_j^t\right| \leq r_o$, then the MC collides with the obstacle[2]. We define C_v^t as the cost of safety violation in slot t, which is represented as follow.

$$C_v^t = \begin{cases} \alpha\left(r_o - \left|l_t - o_j^t\right|\right), & |l_t - o_j^t| \leq r_o; \\ 0, & \left|l_t - o_j^t\right| > r_o, \end{cases} \tag{3}$$

where α denotes the scale factor, which is a constant. In fact, the value of $\alpha\left(r_o - \left|l_t - o_j^t\right|\right)$ indicates the collision

[1] Note that our results still hold when the MC only have the main battery.

[2] For simplicity, we assume that each obstacle has a relatively regular shape.

strength when the MC hits the obstacle. Based on (3), we know that the cost of safety violation is zero when the MC doesn't hit the obstacle, which seems not to give any warning before the collision happens. However we later prove that the cost of safety violation defined in this work can benefit the charging task while preventing the MC from hitting obstacles. Also in practical scenarios, the MC can still maintain the normal operation when slightly hitting the obstacle, which corresponds to the case when C_v^t is a relatively small positive value.

3.2 Wireless Charging Model

We divide each slot t into two phases. The first phase is used for MC's movement, which is illustrated in Subsection 3.1. After the movement in the first phase of slot t, the MC arrives at the new location $l_{t+1} = [x_{t+1}, y_{t+1}]$. Then the MC enters the second phase for delivering the energy to surrounding nodes by omnidirectional WPT techniques. Specifically, we use the wireless charging model developed in [24], which is defined as follow.

$$\begin{cases} p_r = \begin{cases} \frac{\sigma}{(d+\zeta)^2}, & d \le R; \\ 0, & d > R, \end{cases} \\ \sigma = \frac{G_r G_s \eta}{L_p} \left(\frac{\mu}{4\pi}\right)^2 p_c, \end{cases} \tag{4}$$

where p_c and p_r represent the charging power of the MC and the receiving power at the node respectively. We denote d as the distance between the MC and the charged node during the charging process, where $d \le R$ and R is the MC's charging range. We use ζ to adjust the Friis equation for short-distance charging and the remaining parameters defined in (4) are all constants which are co-determined by the node and the MC. We denote s_i^t as the i-th node charged by the MC in slot t, where $i = 1, 2, ..., k_t$ and k_t is the total number of nodes charged in slot t. Note that $k_t = 0$ is feasible when the MC decides not to charge any node or there is not any node within the charging range of the MC in slot t. During the second phase of slot t, multiple nodes can be charged simultaneously as long as they are located within the charging range of the MC. At the end of the second phase, the MC collects feedback signals containing the information of the amount of effective charged energy from all the charged nodes during the second phase. We denote $p_r^t(i)$ as the receiving power at the i-th node s_i^t being charged by the MC in slot t. We denote b_i^t as the residual battery level of node s_i^t at the beginning of slot t. Then the amount of effective[3] charged energy at node s_i^t is represented as

$$e_i^t = \min\left\{\max\left\{b_i^t - g_i^t, 0\right\} + p_r^t(i), B_s\right\} - b_i^t, \tag{5}$$

where g_i^t denotes the amount of energy consumed by node s_i^t in slot t, which is an independent identically distributed Gaussian variable. Also, the residual battery level of node s_i^t evolves over slots as follow.

$$b_i^{t+1} = \min\left\{\max\left\{b_i^t - g_i^t, 0\right\} + p_r^t(i), B_s\right\}. \tag{6}$$

The total amount of effective charged energy by the MC in slot t is $E_t = \sum_{i=1}^{k_t} e_i^t$. Note that since we assume that the length of each slot is one unit time, the amount of energy delivered to node s_i^t equals to the receiving power at node s_i^t in slot t. In addition, we denote b_m^t as the residual battery level of the MC at the beginning of slot t, which evolves over slots as follow.

$$b_m^{t+1} = b_m^t - c_t - p_c^t, \tag{7}$$

where p_c^t denotes the MC's charging power in slot t.

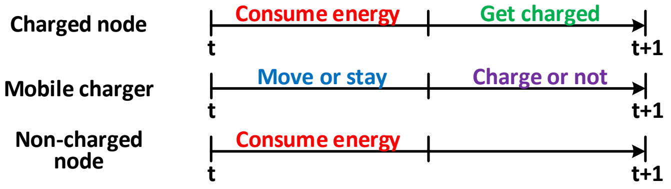


Fig. 2. The system scheduling in one slot.

In Fig. 2, we show the system scheduling in one particular slot. At the beginning of each slot t, each node (including the node to be charged and the node not to be charged) consumes a certain amount of energy due to the node operation (e.g., sensing and communication) and the MC moves to a new location or just stays put according to the control policy π. Next, the system enters the second phase of slot t. During the second phase, the MC charges the nodes within its charging range or decides not to charge any node according to the control policy π. The node to be charged will get charged and the node not to be charged just remains its state of the time when the first phase ends. The real-time residual battery levels of each node and the MC evolve over slots according to (6) and (7) respectively.

4 Problem Formulation

The goal of this work is to maximize the MC's charging utility, i.e., the total amount of effective charged energy by the MC during the time $T_{\mathcal{P}}$. Since $T_{\mathcal{P}}$ is variable which depends on the main battery capacity of the MC and how the MC consumes the onboard energy for movement and charging, it is more appropriate to maximize the average effective charging rate of the MC, which is defined as follow.

$$E_{av} = \frac{1}{T_{\mathcal{P}}} \sum_{t=1}^{T_{\mathcal{P}}} \sum_{i=1}^{k_t} e_i^t, \tag{8}$$

where e_i^t denotes the amount of effective charged energy at the i-th charged node in slot t, which is defined in (5). Based on the network model, we formulate the problem as follow.

$$\max E_{av}, \tag{9}$$

subject to

$$\delta D_{\mathcal{P}} + \sum_{t=1}^{T_{\mathcal{P}}} p_c^t \le B_m, \tag{10}$$

$$|l_{t+1} - l_t| \le \min\left\{\sqrt{2}v_{max}, \frac{b_m^t}{\delta}\right\}, \forall t \in [1, T_{\mathcal{P}}], \tag{11}$$

$$p_c^t \le b_m^t - c_t, \forall t \in [1, T_{\mathcal{P}}]. \tag{12}$$

[3] Here, we use the word "effective" to indicate the real amount of energy delivered to the node by the MC considering the battery overflow.

Constraint (10) states that the total amount of consumed energy on trajectory $\mathcal{P}$ is upper bounded by the MC's main battery capacity. Constraint (11) states that the moving distance in each slot is upper bounded by either the maximum speed or the residual battery level of the MC. Constraint (12) states that the charging power in each slot cannot exceed the residual battery level of the time when the first phase ends. In addition, the MC should try to avoid colliding with both the static and mobile obstacles within the network. In general, we need to jointly design the optimal moving trajectory and charging scheduling to maximize the average effective charging rate based on the energy consumption constraint, the movement constraint, the power allocation constraint and the safety constraint.

Traditional optimization techniques cannot solve (or even explicitly formulate) our problem since the node-side information as well as the obstacle-side information is inaccessible to the MC. For example, the location information of the node and the obstacle is unknown to the MC, although the MC can detect any object within the limited range by using the radar. Since in practical scenarios, the workload of each node may vary over time, which makes the energy consumption rate of each node fluctuate over time. For example, the camera sensor may increase the frame rate when targeting on the moving object. In most cases, the change of the workload of each node is inaccessible to the MC, because the workload usually depends on the stochastic events in the environment. Thus the MC cannot know how the energy consumption rate of each node changes over time. Without the node-side information, the MC cannot decide when and where to charge the nodes within the network. In addition, without the obstacle-side information, the MC cannot maintain a safe moving trajectory.

To tackle the above challenge due to the lack of exact system information, we introduce the RL technique, which enables the MC to learn to optimize the moving trajectory and charging scheduling in a trial-and-error way, by interacting with the environment and tracking feedback signals from nodes and obstacles in real time. In fact, the learning process is complex due to the following reasons. First, the energy consumption rate of each node is time-variant and nodes are randomly distributed within the network. Thus, the real-time residual battery level of each node presents spatial-temporal dynamics within the network. The MC must jointly consider the number of nodes to be charged and their residual battery levels when making the movement and charging decisions in each slot. For example, it is not cost-effective for the MC to travel a long way for charging nodes which all have relatively high residual battery levels or low energy consumption rates, even though the number of nodes to be charged simultaneously is large. Secondly, sometimes the most "profit" nodes, i.e., the nodes which have low residual battery levels or high energy consumption rates, are located near obstacles. Thus there exists the trade-off between maximizing the amount of effective charged energy and minimizing the risk of hitting obstacles, which should be carefully handled during the charging tour. We will show the details in the next section.

5 Joint Trajectory Design and Charging Scheduling with Reinforcement Learning

In this section, we first reformulate the original joint trajectory design and charging scheduling problem as a MDP where the MC makes decisions (i.e., movement and charging) in each slot t based on its state at the beginning of slot t. Then, based on the formulated MDP model, we in detail illustrate how to develop the RL based algorithm to find the optimal policy, which maps the MC's current state to the best decision in each slot. Specifically, at the beginning of each slot t, the MC determines the future movement and charging power allocation based on its current location and residual battery level of the main battery. Under the RL framework, the MC successively makes decisions to maximize the average effective charging rate during the entire charging period $T_{\mathcal{P}}$. At the same time, the MC also needs to consider the safety issue (i.e., collision avoidance) during the charging period $T_{\mathcal{P}}$. To this end, we develop a Soft Actor-Critic based Mobile Security Policy Intervened algorithm (SAC-MSPI), where the distributed exploration based safe training method and SAC based stochastic policy algorithm are incorporated to jointly optimize the safe trajectory design and charging scheduling.

5.1 MDP Formulation

We define the MDP as a five-tuple $\{\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space of the MC, $\mathbf{P}$ denotes the space of state transition probability, $\mathcal{R}$ denotes the reward space and γ denotes the discount factor used to trade off the short-term against long-term cumulative rewards. Based on the MDP, the MC improves its joint trajectory design and charging scheduling via continuously interacting with the environment and maximizing the positive feedback (rewards) from the environment. In particular, we define the MC as the real RL agent. We also define three virtual RL agents, i.e., the *task agent*, the *safety agent* and the *intervention agent*. Each of these three virtual agents represents an independent SAC-based training architecture. The three virtual agents cooperatively control the MC to interact with the environment, which will be introduced later in Section 5.2. We first illustrate each of the MDP elements as follows.

State: We define the MC's real-time location and residual battery level of the main battery as the system state at the beginning of slot t, which is represented as $s_t = \{x_t, y_t, b_m^t\}$. We assume that the state information is completely accessible to the MC since the embedded global positioning system (GPS) module and battery management system can keep monitoring the MC's real-time location and residual battery level. We note that s_t inherits all the historical state information before slot t, i.e., $P\left(s_{t+1}|s_t\right) = P\left(s_{t+1}|s_1, ..., s_t\right)$, based on which the MC makes the next movement and charging decisions. In particular, we assume that the MC is fully charged before the charging task, i.e., we have $b_m^0 = B_m$.

Action: We define the action as the MC's decisions of movement and charging power allocation in each slot t, which is represented as $a_t = \{v_t^x, v_t^y, p_c^t\}$. Specifically, the speed components v_t^x and v_t^y are independently chosen from the range $[-v_{max}, v_{max}]$. In addition, we assume that the charging power p_c^t is fixed over slots during a certain training process, which however may change over different training processes if necessary. We note that the action a_t selected in each slot t must satisfy the constraints defined in (11) and (12).

Reward: We note that the design of the reward function greatly impacts the training effectiveness under the RL framework. In this work, we define the reward function as $r_t = \epsilon \sum_{i=1}^{k_t} e_i^t + (1-\epsilon) k_t$, which is the weighted sum of the amount of effective charged energy and the number of charged nodes in each slot t. The MC receives the reward r_t after performing the action a_t in each slot t. The parameter ϵ is used to adjust the weights. In addition, we assume that the MC will incur a certain penalty, i.e., the cost of safety violation defined in (3), when colliding with any obstacle during the training process. Thus the goal of the developed training algorithm is to maximize the average cumulative rewards and minimize the average cumulative costs of safety violation at the same time. We note that the cost of safety violation is individually developed, rather than being incorporated into the reward function. Because the method of combining the reward and the cost as a single feedback signal may prevent the MC from exploring better charging locations near obstacles for reducing the risk of hitting obstacles.

State transition probability: The state transition probability $P(s_{t+1}, r_t | s_t, a_t)$ defines the system dynamics dependent only on the preceding state and action. That is, for particular values of these two random variables $s_{t+1} \in \mathcal{S}$ and $r_t \in \mathcal{R}$, there is a probability of those values occurring in slot t, given particular values of the preceding state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. Since both of the state space and action space are continuous and infinite, the space of the state transition probability $\mathbf{P}$ is continuous and infinite, which thus cannot be explicitly modeled. Thus in this work, we apply the model-free RL technique to enable the MC to learn to optimize the joint trajectory design and charging scheduling by interacting with the environment without the explicit model of state transition probability.

5.2 The Framework of SAC-MSPI

Motivation: Since the agent needs to explore and learn the unknown environment in a trial-and-error way to improve its movement and charging strategy, which may cause the damage to the agent when some aggressive actions are tried. For our case, the MC needs to consider not only the maximization of charging utility, but also the safety of movement within the network due to the existence of both static and mobile obstacles. Thus keeping the MC safe while performing the charging task is extremely important, especially when the MC is deployed in unknown and complex environments. In this work, we propose the SAC-MSPI, which combines the distributed exploration based safe training method and SAC based stochastic policy algorithm to optimize the joint trajectory design and charging scheduling. In fact, the proposed SAC-MSPI improves traditional safe RL frameworks by not sacrificing the charging utility for the safety of movement. Specifically, to guarantee the safety of the agent during the agent-environment interaction, traditional safe RL techniques either use the prior knowledge of some risky states to restrict the action selection or incorporate the cost of safety violation into the reward function, which usually sacrifices the task utility for the safe exploration in unknown environments.

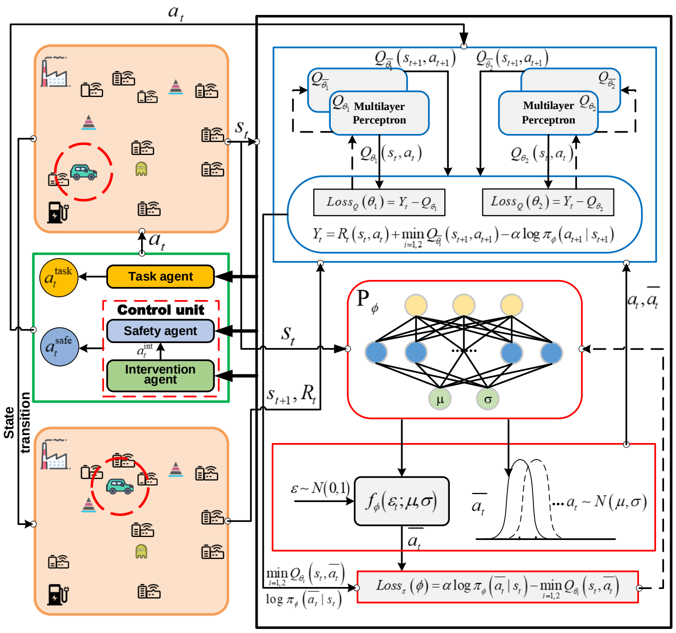

a_t
s_t
a_t
Task agent
Control unit
Safety agent
Intervention agent
a_t^{task}
a_t^{safe}
a_t^{int}
State transition
s_t
s_{t+1}, R_t
$Q_{\bar{\theta}_1}(s_{t+1}, a_{t+1})$
$Q_{\bar{\theta}_2}(s_{t+1}, a_{t+1})$
$Q_{\bar{\theta}_1}$
$Q_{\bar{\theta}_2}$
Q_{θ_1} Multilayer Perceptron
Multilayer Perceptron Q_{θ_2}
$Q_{\theta_1}(s_t, a_t)$
$Q_{\theta_2}(s_t, a_t)$
$Loss_Q(\theta_1) = Y_t - Q_{\theta_1}$
$Loss_Q(\theta_2) = Y_t - Q_{\theta_2}$
$Y_t = R_t(s_t, a_t) + \min_{i=1,2} Q_{\bar{\theta}_i}(s_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(a_{t+1} | s_{t+1})$
P_ϕ
μ
σ
$a_t, \bar{a}_t$
$\varepsilon \sim N(0,1)$
$f_\phi(\varepsilon_t; \mu, \sigma)$
$\bar{a}_t$
$a_t \sim N(\mu, \sigma)$
$\min_{i=1,2} Q_{\theta_i}(s_t, \bar{a}_t)$
$\log \pi_\phi(\bar{a}_t | s_t)$
$Loss_\pi(\phi) = \alpha \log \pi_\phi(\bar{a}_t | s_t) - \min_{i=1,2} Q_{\theta_i}(s_t, \bar{a}_t)$

Fig. 3. The system framework of SAC-MSPI.

The core architecture of SAC-MSPI includes three virtual RL agents, i.e., the task agent, the safety agent and the intervention agent. At any time, either the task agent or the safety agent can directly control the MC via their respective learned policies. The intervention agent only decides when the safety agent takes control of the MC, rather than directly control the MC. In particular, we define the task policy, the security policy and the intervention policy as the policies learned by the task agent, the safety agent and the intervention agent respectively. Each agent independently and sequentially makes action decisions based on the system state defined in Section 5.1 as well as the prediction of other agents' future actions. In the following, we first introduce how the task agent and the safety agent alternately take control of the MC based on the decision made by the intervention agent.

Security intervened policy switching mechanism: To handle the tradeoff between maximizing the charging utility and minimizing the cost of safety violation, we introduce the intervention agent to decide the appropriate time when the safety agent should take control of the MC, since the task agent controls the MC most of the time during the charging task for maximizing the charging utility. Once the safety agent is chosen to control the MC and execute the corresponding security policy, it means that the MC has already hit the obstacle or the collision may happen in the near future. Thus the intervention agent needs to learn and improve its intervention strategy, i.e., knowing the best time to let the MC execute the security policy instead of the task policy, based on the MC's state and the action performed by the safety agent. In this way, the MC may avoid hitting obstacles during the charging task. In Fig. 3, we show the system framework of SAC-MSPI, which consists of three virtual RL agents and a control unit. Based on each system state s_t, both of the task agent and the safety agent will make their action decisions, which are denoted as $a_t^{\text{task}} \in \mathcal{A}$ and $a_t^{\text{safe}} \in \mathcal{A}^{\text{safe}} \subseteq \mathcal{A}$ respectively. Then, the control unit G decides whether to adopt the action decision a_t^{safe} instead of a_t^{task}. With the control unit G, the MC only executes action a_t^{safe} at some states which are assumed to be safety-violated (or potentially) by the intervention agent. As for all the other states, the MC devotes to executing and optimizing the action a_t^{task} to maximize the average effective charging rate. Thus, the objective of jointly maximizing the charging utility and minimizing the cost of safety violation is decoupled into two separate objectives, which are assigned to the task agent and the safety agent respectively. We note that the reward loss caused by trading off the MC's charging utility and safety of movement can be reduced by the above decoupling method, which is proved in the evaluation section.

Task agent: The objective of the task agent is to maximize the expected accumulated rewards, i.e., the total amount of effective charged energy by the MC during each training episode. The task agent can freely choose and execute actions by following the stochastic policy at any state which is not intervened by the safety agent. The reward function of the task agent is defined as

$$\begin{aligned} R_t^{\text{task}}\left(s_t, a_t^{\text{task}}, a_t^{\text{safe}}\right) = r_t\left(s_t, a_t^{\text{task}}\right)\left(1-\mathbf{I}_{\text{safe}}^t\right) \\ + r_t\left(s_t, a_t^{\text{safe}}\right)\mathbf{I}_{\text{safe}}^t, \end{aligned} \tag{13}$$

where $r_t(\cdot)$ is defined in Section 5.1. We denote $\mathbf{I}_{\text{safe}}^t$ as the action decision made by the intervention agent in slot t, where $\mathbf{I}_{\text{safe}}^t \in \{0, 1\}$. When $\mathbf{I}_{\text{safe}}^t = 1$, the MC executes the safety action a_t^{safe} determined by the safety agent. The task agent will receive the reward $r_t\left(s_t, a_t^{\text{safe}}\right)$. Otherwise, the MC executes the task action a_t^{task} and the task agent receives the reward $r_t\left(s_t, a_t^{\text{task}}\right)$. We note that at any time, the reward obtained by the task agent only depends on the charging utility, but not including the cost of safety violation. Thus, the task agent seeks to maximize the expected accumulated rewards when starting in any state $s \in \mathcal{S}$, which is represented as

$$v_{\pi,(\pi^{\text{safe}},G)}^{\text{task}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t^{\text{task}}\left(s_t, a_t^{\text{task}}, a_t^{\text{safe}}\right) \middle| s_t = s\right], \tag{14}$$

where $a_t^{\text{task}} \sim \pi^{\text{task}}(\cdot|s_t)$ is the action decision made based on the policy π^{task} of the task agent in slot t, and $a_t^{\text{safe}} \sim \pi^{\text{safe}}(\cdot|s_t)$ is the action decision made based on the policy π^{safe} of the safety agent in slot t.

Safety agent: We previously stated that normally the task agent controls the MC to move and charge nodes, unless when the intervention agent decides to let the safety agent take control of the MC, which generates the cost of safety violation as well as the intervention cost. The objective of the safety agent is to minimize the expected accumulated costs of safety violation and intervention, which thus reduces the number of states of safety violation during both the training and working episodes. Without loss of generality, we define the reward function of the safety agent as

$$\begin{aligned} R_t^{\text{safe}}\left(s_t, a_t^{\text{task}}, a_t^{\text{safe}}\right) = -\Big[C_v^t\left(s_t, a_t^{\text{task}}\right)\left(1-\mathbf{I}_{\text{safe}}^t\right) + \\ C_v^t\left(s_t, a_t^{\text{safe}}\right)\mathbf{I}_{\text{safe}}^t\Big] - c_{\text{int}}\mathbf{I}_{\text{safe}}^t, \end{aligned} \tag{15}$$

where $C_v^t(\cdot)$ is defined in (3) and c_{int} is a constant which denotes the intervention cost of each event when the intervention agent decides to let the safety agent take control of the MC. Since the job of the intervention agent is to guarantee the safety of the MC by intermittently intervening the operation of the task agent, it is important for the intervention agent to intervene at the appropriate time when the MC really needs to be warned of the safety violation during the charging task. Otherwise, the MC may finish the charging task safely, which however may reduce the chance of obtaining the larger charging utility. Thus, under the control of the intervention agent, the safety agent seeks to maximize the expected accumulated rewards when starting in any state $s \in \mathcal{S}$, which is represented as

$$v_{\pi,(\pi^{\text{safe}},G)}^{\text{safe}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t^{\text{safe}}\left(s_t, a_t^{\text{task}}, a_t^{\text{safe}}\right) \middle| s_t = s\right]. \tag{16}$$

5.3 Design and Implementation of SAC-MSPI Algorithm

Architecture: The SAC-MSPI is designed based on the SAC algorithm. That is, each of the three agents (i.e., the task agent, the safety agent and the intervention agent) is trained and controlled by an independent SAC module. The core of SAC-MSPI is the adaptive game between the task agent and the safety agent controlled by the intervention agent. At any system state, the task agent and the safety agent coordinate to execute actions and learn to predict future states, for maximizing the average effective charging rate while minimizing the number of unsafe states. The SAC-MSPI algorithm includes three main parts, i.e., the data collection, the control of policy switching and the policy update, which are shown in Fig. 3 and Algorithm 1. In general, at each state s_t the MC executes the action a_t^{task} or a_t^{safe} outputted by the policy switching module and the resulting data (i.e., the previous state, the executed action, the next state and the obtained reward) will be stored in the data buffer. When the data buffer has sufficient data as the agent-environment interaction proceeds, the three SAC modules start to train the model synchronously by sampling data from the data buffer.

Work flow: The whole system operates under the control of Algorithm 1. Specifically, we first initialize three data buffers $\mathcal{D}_{\text{task}}$, $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{int}}$. We also initialize the hyper-parameters of each SAC module, including ϕ, θ_i and $\bar{\theta}_i$, where $i \in \{1, 2\}$. Starting in any state s_t, the policy switching module outputs the specific action a_t^{task} or a_t^{safe}, which is then executed by the MC. Then the system enters the next state s_{t+1} and each agent receives the corresponding reward R_t^{task} or R_t^{safe}. After a few number of iterations, the system generates sufficient number of data samples, which will be stored in the data buffer. In particular, both of data buffers $\mathcal{D}_{\text{task}}$ and $\mathcal{D}_{\text{safe}}$ store the same triad $\left\{s_t, a_t^{\text{task}}/a_t^{\text{safe}}, s_{t+1}\right\}$. The rewards R_t^{task} and R_t^{safe} are stored in $\mathcal{D}_{\text{task}}$ and $\mathcal{D}_{\text{safe}}$ respectively. In addition, the quadruple $\left\{s_t, 0/1, s_{t+1}, R_t^{\text{safe}}\right\}$

Algorithm 1 SAC-MSPI algorithm

```
Initialize data buffers $\mathcal{D}_{\text{task}} = \mathcal{D}_{\text{safe}} = \mathcal{D}_{\text{int}} = \emptyset$
Initialize hyper-parameters $\theta_1, \theta_2, \phi, \bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ for each SAC module
Total Steps: Steps $= 1$
for $N_{\text{episodes}}$ do
    State $s_0$
    while $b_m^t > 0$ do
        Sample a task action $a_t^{\text{task}} \sim \pi^{\text{task}}(\cdot|s_t)$, a safe action $a_t^{\text{safe}} \sim \pi^{\text{safe}}(\cdot|s_t)$, and an intervention action $a_t^{\text{int}} \sim g(\cdot|s_t)$
        if $a_t^{\text{int}} > 0$ then
            Execute safe action $a_t^{\text{safe}}$ so $s_{t+1} \sim P(\cdot|a_t^{\text{safe}}, s_t)$
            Set $a_t = a_t^{\text{safe}}$, $a_t^{\text{int}} = 1$
        else
            Execute task action $a_t^{\text{task}}$ so $s_{t+1} \sim P(\cdot|a_t^{\text{task}}, s_t)$
            Set $a_t = a_t^{\text{task}}$, $a_t^{\text{int}} = 0$
        end if
        Receive reward $R_t^{\text{task}}(s_t, a_t)$, $R_t^{\text{safe}}(s_t, a_t)$ and $R_t^{\text{int}}(s_t, a_t^{\text{int}}) = R_t^{\text{safe}}(s_t, a_t)$
        Add the sample $(s_t, a_t, s_{t+1}, R_t^{\text{task}})$ to $\mathcal{D}_{\text{task}}$, the sample $(s_t, a_t, s_{t+1}, R_t^{\text{safe}})$ to $\mathcal{D}_{\text{safe}}$, the sample $(s_t, a_t^{\text{int}}, s_{t+1}, R_t^{\text{safe}})$ to $\mathcal{D}_{\text{int}}$
        if Steps $\geq$ StepsLearn then
            For each of the three agents do:
            Sample a batch of $\{(s_t, a_t, R_t, s_{t+1})\}_{t=1,\ldots,\mathcal{B}}$ from the corresponding data buffer $\mathcal{D}$
            Update $\mathcal{Q}_{\theta_i}$ by minimizing the loss function defined in Equation (18), where $i = 1, 2$
            Sample $\bar{a}_t$ by using the reparameterization trick
            Update $\mathcal{P}_\phi$ by minimizing the loss function defined in Equation (21)
            Update $\alpha$
            Update $\bar{\theta}_i$: $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i, \forall i \in \{1, 2\}$
        end if
        Steps = Steps + 1
    end while
end for
```

will be stored in $\mathcal{D}_{\text{int}}$. As the training proceeds, each data buffer accumulates data samples and each agent collects data samples from its corresponding data buffer. The training during each episode ends when the main battery of the MC is exhausted. To achieve the precise policy switching between the task agent and the safety agent, we convert the continuous action of the intervention agent into two binary decision variables, i.e., $a_t^{\text{int}} \sim g(\cdot|s_t) \in \{0, 1\}$. When the SAC module of the intervention agent outputs the action value larger than zero, then the MC executes the safe action. Otherwise, the MC executes the task action. Each agent is trained by using the policy gradient algorithm SAC and the off-policy training method is applied to reduce the complexity of sampling. Each SAC module updates its hyper-parameters at intervals of a certain number of training steps based on the requirement of the policy learning.

SAC module: Soft Actor-Critic (SAC) is a RL algorithm that uses the off-policy method to optimize the stochastic policy. The actor tries to maximize the expected reward and the expected entropy of the learned policy simultaneously, which thus enables the agent to better handle the disturbance in practical scenarios while guaranteeing the maximum achieved task utility. In particular, when multiple actions indicate the same large future accumulated rewards, the agent will randomly execute one of these actions with equal probability. Given a particular policy π, we define the sum of the expected reward and the expected entropy of π as

$$\mathcal{J}(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, a_t)\sim\rho_\pi}\left[R_t(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))\right], \quad (17)$$

where α is the entropy coefficient that can be learned automatically [50]. We denote ρ_π as the state-action trajectory distribution induced by policy π and $\mathcal{H}$ is the entropy of each visited state. The SAC module targets on maximizing the objective function $\mathcal{J}(\pi)$.

To maximize $\mathcal{J}(\pi)$, the policy π needs to be improved by iteration, i.e., we iteratively evaluate the newly updated policy using the soft Q-function and perform the policy improvement. Since both the state and action spaces of the MC are continuous, we develop three function approximators (neural networks) including two Q-networks ($\mathcal{Q}_{\theta_1}$, $\mathcal{Q}_{\theta_2}$) and a P-network ($\mathcal{P}_\phi$), to parameterize the state-action value function $Q_{\theta_i}(s_t, a_t)$[4] and the policy function $\pi_\phi(a_t|s_t)$ respectively. Both of the two Q-networks and the P-network are alternately trained with the stochastic gradient descent method.

To alleviate the problem of overestimating the Q-value, each Q-network $\mathcal{Q}_{\theta_i}$ is accompanied by a target Q-network $\mathcal{Q}_{\bar{\theta}_i}$, where $i = 1, 2$. Each Q-network $\mathcal{Q}_{\theta_i}$ is trained and updated by minimizing the soft Bellman residual defined in (18), where the V-function is defined in (19).

$$\begin{aligned} L_Q(\theta_i) = \mathbb{E}_{(s_t, a_t)\sim D}\Bigg[\frac{1}{2}\Big(Q_{\theta_i}(s_t, a_t) - \big(R_t(s_t, a_t) + \\ \gamma\mathbb{E}_{s_{t+1}\sim\rho_\pi}\left[V_{\bar{\theta}_i}(s_{t+1})\right]\big)\Big)^2\Bigg], \end{aligned} \quad (18)$$

where

$$\begin{aligned} V_{\bar{\theta}_i}(s_{t+1}) = \mathbb{E}_{a_{t+1}\sim\pi_\phi}\Bigg[\min_{i=1,2} Q_{\bar{\theta}_i}(s_{t+1}, a_{t+1}) - \\ \log\pi_\phi(a_{t+1}|s_{t+1})\Bigg]. \end{aligned} \quad (19)$$

Each target Q-network $\mathcal{Q}_{\bar{\theta}_i}$ is asynchronously updated by replacing $\bar{\theta}_i$ with an exponential moving average of θ_i.

The policy π is updated based on the exponential of the new Q-function, which ensures that the policy is improved according to the soft Q-value. In other words, this method aligns the distribution of the newly updated policy's actions with the distribution of new soft Q-values. Furthermore, since the output of the P-network consists of both the mean μ and the standard deviation σ, we cannot calculate the gradient by sampling action a_t directly from the resulting normal distribution $\mathcal{N}(\mu, \sigma)$. Instead, we perform the gradient back-propagation by re-sampling action $\bar{a}_t$ from an approximate distribution generated based on $\mathcal{N}(\mu, \sigma)$ and

[4] For ease of illustration when describing the SAC module, we use a_t as the unified representation of different agents' actions. The same is true with respect to other related elements such as the policy π and the reward R_t.

then update the hyper-parameter ϕ of the P-network. The loss function of the P-network is defined as

$$\begin{aligned}
L_{\pi}(\phi) &= D_{KL}\left(\pi_{\phi}^{\text{new}}\left(\cdot|s_t\right)\,\middle|\,\frac{\exp\left(\frac{1}{\alpha}Q^{\pi_{\text{old}}}\left(s_t,\cdot\right)\right)}{Z^{\pi_{\text{old}}}\left(s_t\right)}\right)\\
&= \mathbb{E}_{s_t\sim\mathbb{D},a_t\sim\pi_{\phi}}\left[\log\left(\frac{Z\left(s_t\right)\pi_{\phi}\left(a_t|s_t\right)}{\exp\left(\frac{1}{\alpha}Q\left(s_t,a_t\right)\right)}\right)\right]\\
&= \mathbb{E}_{s_t\sim\mathbb{D},a_t\sim\pi_{\phi}}\left[\log\pi_{\phi}(a_t|s_t)-\frac{1}{\alpha}Q(s_t,a_t)+\log Z(s_t)\right],
\end{aligned} \tag{20}$$

Where $\log Z(s_t)$ is a constant coefficient which can be ignored. Thus Equation (20) is reformulated as

$$\mathbb{E}_{s_t\sim\mathbb{D}}\left[\mathbb{E}_{\bar{a}_t\sim\pi_{\phi}}\left[\alpha\log\left(\pi_{\phi}\left(\bar{a}_t|s_t\right)\right)-\min_{i=1,2}Q_{\theta_i}\left(s_t,\bar{a}_t\right)\right]\right]. \tag{21}$$

6 Evaluation

In this section, we first evaluate the performance of SAC-MSPI including the charging utility and the capability of collision-avoidance, by comparing SAC-MSPI with existing main unconstrained or constrained RL algorithms. Secondly, we study the impact of each system parameter on the training performance such as the convergence speed and the expected cumulative rewards. Thirdly, we compare the SAC-MSPI with several traditional baseline algorithms such as greedy and random algorithms. Finally, we show the MC's learned trajectory in different scenarios, providing the visualized analysis of the effectiveness and robustness of SAC-MSPI.

we consider a 6×6 2D coordinate system where (x,y) is denoted as the location of any object in the 2D plane. We assume that nodes and static obstacles are randomly distributed within the 2D plane. Mobile obstacles may move randomly within the 2D plane. The charging station (starting point) is located at $(0,0)$. The MC's moving direction ranges in $[0,360°]$. The battery capacity of each node is $B_s=8$. We conduct the evaluation based on the following six experiment scenarios.

Scenario 1: The initial battery level and the moving cost per unit distance of the MC are $B_m=200$ and $\delta=0.2$ respectively. The charging power and the charging range of the MC are $p_c=2$ and $R=0.2$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.3]$. There is one static obstacle deployed at $(0.9,1.8)$.

Scenario 2: The initial battery level and the moving cost per unit distance of the MC are $B_m=200$ and $\delta=1$ respectively. The charging power and the charging range of the MC are $p_c=2$ and $R=0.2$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.3]$. There is one static obstacle deployed at $(0.9,1.8)$.

Scenario 3: The initial battery level and the moving cost per unit distance of the MC are $B_m=100$ and $\delta=0.2$ respectively. The charging power and the charging range of the MC are $p_c=2$ and $R=0.2$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.3]$. There is one static obstacle deployed at $(0.9,1.8)$.

Scenario 4: The initial battery level and the moving cost per unit distance of the MC are $B_m=200$ and $\delta=0.2$ respectively. The charging power and the charging range of the MC are $p_c=4$ and $R=0.3$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.3]$. There is one static obstacle deployed at $(0.9,1.8)$.

Scenario 5: The initial battery level and the moving cost per unit distance of the MC are $B_m=80$ and $\delta=1.5$ respectively. The charging power and the charging range of the MC are $p_c=5$ and $R=0.4$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.8]$. There is one static obstacle deployed at any of the following three locations $(1.5,2.5)$, $(2.1,1.5)$ and $(2.5,3.5)$.

Scenario 6: The initial battery level and the moving cost per unit distance of the MC are $B_m=80$ and $\delta=1.5$ respectively. The charging power and the charging range of the MC are $p_c=5$ and $R=0.4$ respectively. The MC's speed components v_t^x and v_t^y both range in $[0,0.8]$. There are four static obstacles deployed at $(1.4,1)$, $(1.5,2)$, $(2.5,3.5)$ and $(3.5,3.8)$ respectively. A mobile obstacle starts at $(1,0)$ and randomly changes its location every two slots.

In Scenarios 1-4, we assume that the initial battery level of each node is independently and identically sampled from a normal distribution $\mathcal{N}_{\text{ini}}\left(u_{\text{ini}}=7,\sigma_{\text{ini}}=0.5\right)$. Also, we assume that the amount of energy consumed in each slot at each node is independently and identically sampled from a normal distribution $\mathcal{N}_{\text{con}}\left(u_{\text{con}}=0.04,\sigma_{\text{con}}=0.08\right)$. In Scenarios 5-6, we assume that the initial battery levels of different nodes may be sampled from different normal distributions and the same is true with respect to the amount of energy consumed in each slot at different nodes. The detailed parameter settings will be presented in Section 6.4.

6.1 Comparison of SAC-MSPI with Existing Unconstrained or Constrained RL Algorithms

To evaluate the performance of the proposed SAC-MSPI algorithm with respect to the charging utility and the capability of collision-avoidance, we compare SAC-MSPI with existing main RL algorithms by evaluating the following three performance metrics [51], i.e., the average episodic return (the average cumulative rewards obtained by an average of 64 tests executed every 2000 training steps), the average episodic costs (the average cumulative violation costs obtained by an average of 64 tests executed every 2000 training steps) and the cost rates (the average cumulative costs obtained by an average of every 2000 training steps during the entire training process). We compare SAC-MSPI with the unconstrained RL algorithms including soft actor-critic (SAC) [52], trust region policy optimization (TRPO) [53] and proximal policy optimization (PPO) [54]. Specifically, we incorporate the cost of safety violation into the reward function when applying those unconstrained RL algorithms. We compare SAC-MSPI with the constrained RL algorithms including the TRPO-Lagrangian and PPO-Lagrangian, which employ the adaptive penalty coefficient to enforce constraints and solve $\max_{\theta}\min_{\lambda}\left[f\left(\theta\right)-\lambda g\left(\theta\right)\right]$ by gradient ascent on θ and gradient descent on λ, where f and g represent the objective function and the constraint function respectively. We also compare SAC-MSPI with the constrained policy optimization (CPO) algorithm [55]. CPO is the constrained form of TRPO, which recalculates the penalty coefficient for each update.

We compare the proposed SAC-MSPI algorithm with the above benchmark RL algorithms based on Scenario 1. Fig. 4

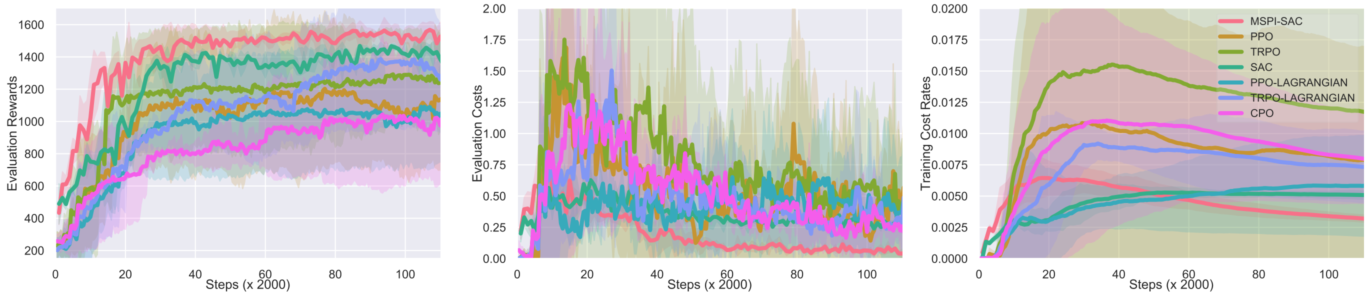


Fig. 4. The comparison of SAC-MSPI with existing unconstrained and constrained RL algorithms.

demonstrates that SAC-MSPI achieves the highest cumulative rewards, the lowest cumulative costs of safety violation and the lowest cost rates, compared with all the benchmark RL algorithms. During the training process, we conduct 64 tests every 2000 training steps and three random seeds are used for each evaluated algorithm. The proposed SAC-MSPI algorithm maintains the performance stability across different random seeds, which also proves its robustness.

6.2 The Impact of System Parameters on the Training Performance

In this section, we investigate the impact of different system parameters (e.g., the MC's charging power, the moving cost per unit distance and the main battery capacity) on the training performance such as the convergence and the maximum achieved cumulative rewards by the proposed algorithm.

Charging power: When the MC uses the large charging power to charge nodes with relatively high battery levels, a large amount of energy will be wasted due to the battery overflow. On the contrary, when the MC uses the small charging power to charge nodes with relatively low battery levels, the MC may repeatedly charge these nodes to satisfy their charging demands, which however may degrade the charging utility of other nodes. In Fig. 5 and Fig. 6, the blue curve is generated based on Scenario 1 and the purple curve is generated based on Scenario 4. The result shows that increasing the charging power may not increase the cumulative rewards (actually decrease the cumulative rewards in this case). Because the real-time residual battery level of each node presents spatial-temporal dynamics within the network and sometimes the high charging power may waste a large amount of energy in some charging locations. Also, increasing the charging power may substantially reduce the running time of the MC due to its finite battery capacity, which is proved in Fig. 6 where the purple curve converges to a relatively low value compared with the blue curve. By reducing the running time of the MC, there will be more nodes losing the chance of getting charged, which thus degrades the average effective charging rate, i.e., the charging utility.

Moving cost per unit distance: Since the energy stored in the MC's main battery is used for both charging nodes and movement, the moving cost per unit distance as well as the total length of the moving trajectory greatly affects the achievable cumulative rewards. We try to reduce the energy consumption of the movement, which thus can

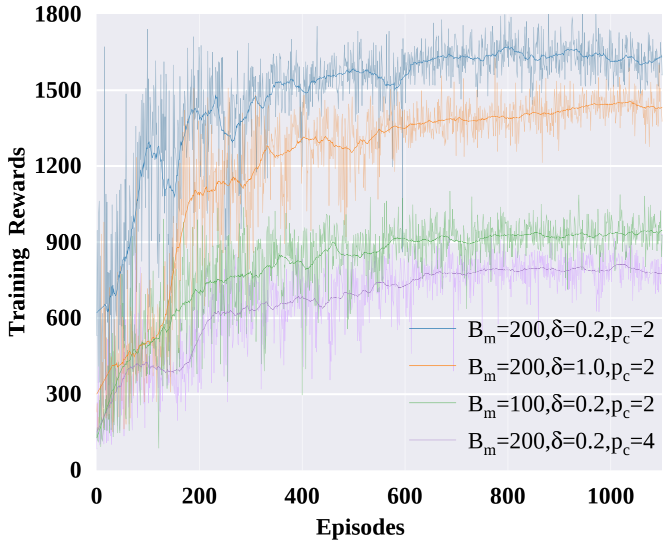


Fig. 5. The reward changes with the training episode in different experiment scenarios.

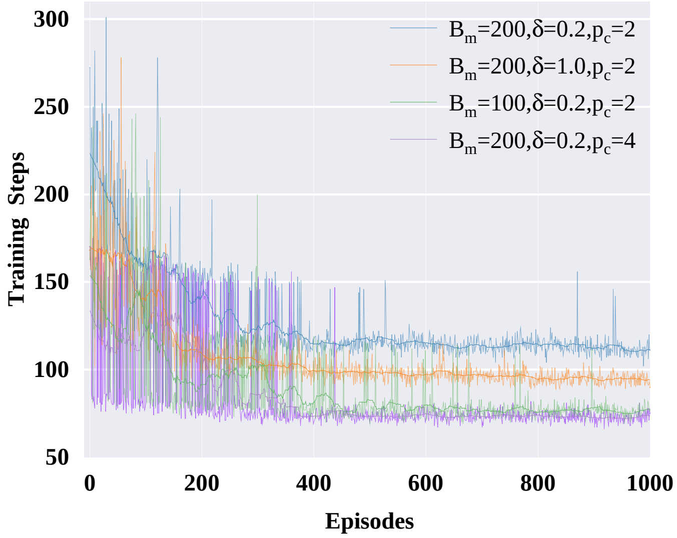


Fig. 6. The convergence of the SAC-MSPI algorithm in different experiment scenarios.

maximize the total amount of energy used for charging nodes. Although the low moving cost per unit distance and the short moving trajectory can both save the energy used for MC's movement, sometimes it is necessary for the MC to extend its moving trajectory for maximizing the total amount of effective charged energy by charging more nodes. In Fig. 5 and Fig. 6, the orange curve is generated based on Scenario 2 where the moving cost per unit distance of the MC is relatively high. Compared with the orange curve, we can see that the blue curve converges to a higher value (higher cumulative rewards) since the MC employs the lower moving cost per unit distance in Scenario 1. In Fig. 6, we can see that the running time of the MC is reduced due to the higher moving cost per unit distance, by comparing the blue curve and the orange curve.

Main battery capacity: The MC's main battery capacity also affects the achievable cumulative rewards and the running time of the MC. In Fig. 5 and Fig. 6, the green curve is generated based on Scenario 3 where the main battery capacity of the MC is relatively small. Compared with the green curve, we can see that the higher main battery capacity of the MC results in the higher cumulative rewards and larger running time, since the MC can carry the larger amount of energy from the starting point.

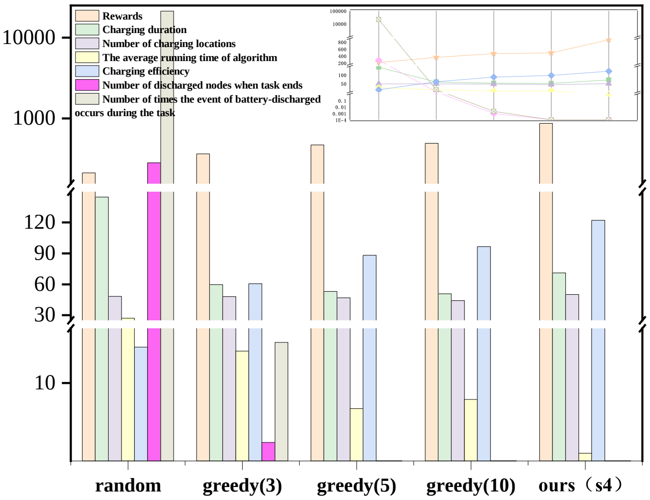


Fig. 7. The comparison of the SAC-MSPI with random and greedy algorithms.

6.3 Comparison of SAC-MSPI with Traditional Baseline Algorithms

In this section, we compare the proposed SAC-MSPI algorithm with two traditional baseline algorithms, i.e., the random algorithm and the greedy algorithm. Note that both of the random and greedy algorithms are designed based on the fact that the node-side information as well as the obstacle-side information is accessible to the MC before the charging task. When designing the random algorithm, we assume that at the beginning of each slot t, the MC decides to move to the next location which is randomly chosen within a square area centered at the MC's current location. The side-length of this square area is 0.6. The MC charges the nodes within its charging range once it arrives at the new location. As for the greedy algorithm, we assume that at the beginning of each slot t, the MC first randomly chooses multiple candidate next-to-go locations within a square area centered at the MC's current location (The side-length of this square area is 0.6). Then we calculate the respective amount of effective charged energy when the MC charges nodes at each of those candidate locations. Finally, the MC decides to move to one of the candidate locations with the maximum achievable amount of effective charged energy. The MC charges the nodes within its charging range once it arrives at the new location. In particular, the number of candidate locations is set as 3, 5 and 10. Thus, the corresponding greedy algorithms are defined as greedy(3), greedy(5) and greedy(10) respectively. We also assume that there exist no obstacles within the network when running either the random algorithm or the greedy algorithm. Even under this circumstance, the SAC-MSPI algorithm outperforms both the random and greedy algorithms with respect to the charging utility, which is proved in Fig. 7.

We compare the SAC-MSPI algorithm with the random and greedy algorithm based on Scenario 4. Figure 7 shows the performance of each of the three evaluated algorithm with respect to the cumulative rewards, the duration of the charging task, the number of charging locations, the average running time of the algorithm, the charging efficiency, the number of nodes with discharged battery when the charging task ends and the number of times the event of battery-discharged has occurred during the charging task. Our algorithm outperforms the random and greedy algorithms in terms of the charging reward and charging efficiency. When each node maintains its normal operation, i.e., the battery level never drops to zero, during the entire charging task, we can see that the SAC-MSPI algorithm still achieves the higher charging efficiency and the smaller running time of algorithm, compared with the greedy(5) and greedy(10).

6.4 Analysis of the MC's Learned Moving Trajectory in Different Experiment Scenarios

In this section, we mainly study the MC's learned trajectory in different experiment scenarios, since the charging utility as well as the movement safety of the MC strongly depends on its learned trajectory. Specifically, we investigate the impact of different network settings on the learned trajectory, including the initial battery levels of nodes, the energy consumption rates of nodes and the distributed density of nodes within the network. To ease of presentation, in Figs. 8-12, we use the diamond to represent each of the nodes with higher energy consumption rates on average. The pentagram is used to represent each of the nodes with lower energy consumption rates on average. The nodes in different colors have various initial battery levels on average. In addition, we also evaluate the effectiveness of the proposed SAC-MSPI algorithm in the following two cases: 1) There are multiple static obstacles within the network; 2) There is one mobile obstacle that moves randomly within the network.

Initial battery levels of nodes: Based on Scenario 5, each node has the same average energy consumption rate. The nodes in different colors have different average initial battery levels. Specifically, the nodes in purple, brown,

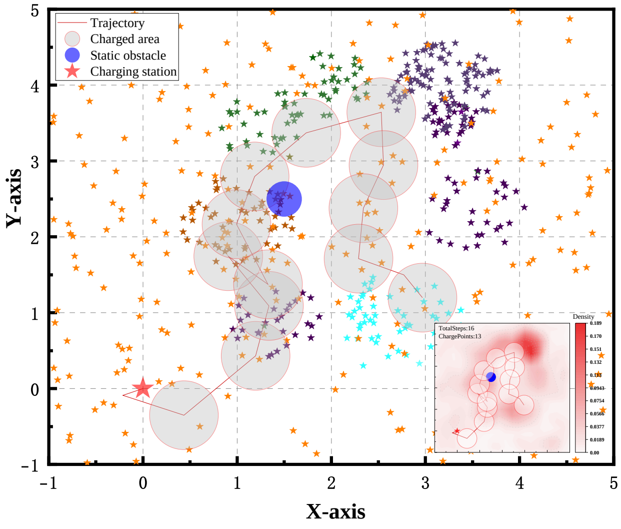


Fig. 8. The MC's learned trajectory when nodes in different regions have various average initial battery levels.

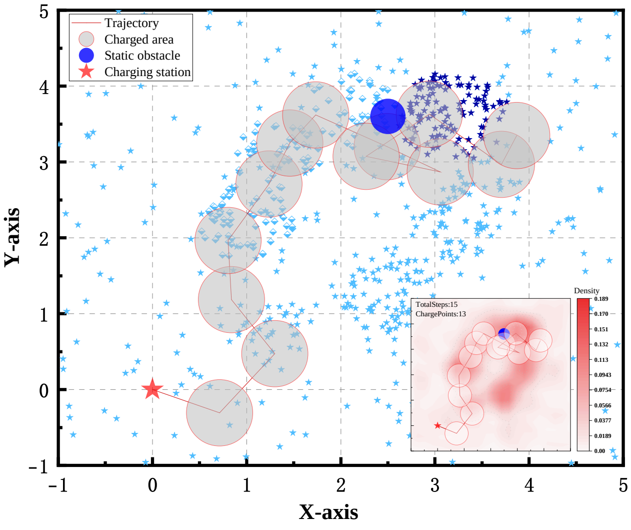


Fig. 10. The MC's learned trajectory when nodes in different regions have various average energy consumption rates.

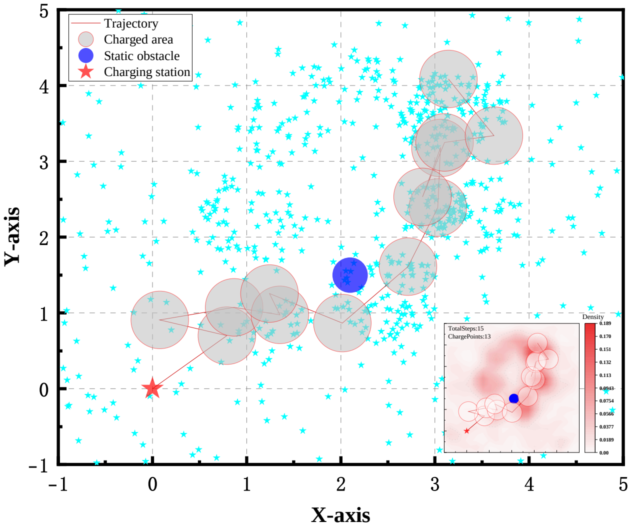


Fig. 9. The MC's learned trajectory when the distributed density of nodes changes over regions.

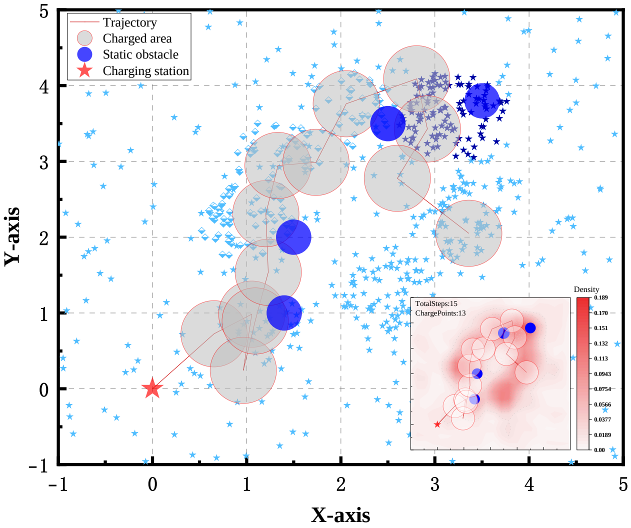


Fig. 11. The MC's learned trajectory when there are multiple static obstacles.

green, orange and blue have the average initial battery levels of 6, 3, 4, 8 and 7 respectively. Fig. 8 shows that the MC starts from the charging station and then successively passes through the regions of nodes in purple, brown, green, purple and blue. Results show that the MC first moves to the region where the nodes have relatively low initial battery levels, since charging these nodes results in the larger amount of effective charged energy compared with the nodes with relatively high initial battery levels. Also, giving the charging priority to the nodes with relatively low initial battery levels can reduce the number of times an event of battery-discharged has occurred during the charging task. The small figure in the bottom-right corner of Fig. 8 shows the distributed density of nodes within the network as well as the MC's learned trajectory. Although the distributed density of nodes located in the top-right area of the network is high, which indicates a large number of nodes that can be simultaneously charged by the MC at each charging location, the simulation result shows that the MC doesn't spend much time visiting that area since the nodes in the top-right area have relatively high initial battery levels. This result proves the intelligence of the proposed SAC-MSPI algorithm.

Distributed density of nodes: Fig. 9 is generated based on Scenario 5 where the initial battery level of each node is independently and identically sampled from the same Gaussian distribution. Also, the amount of energy consumed in each slot at each node is independently and identically sampled from the same Gaussian distribution, i.e., each node has the same average energy consumption rate. The

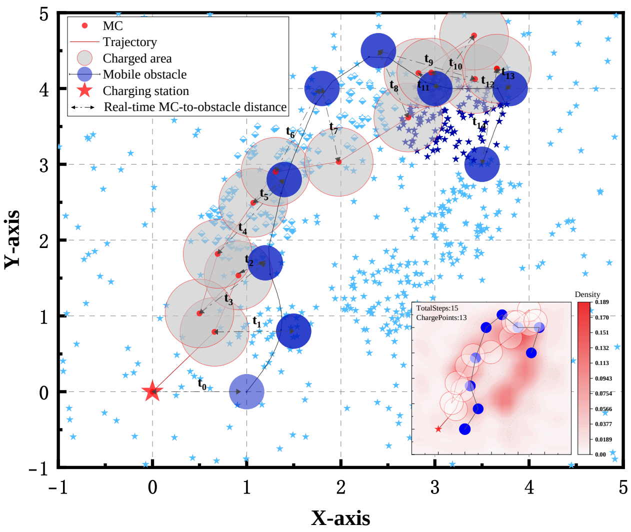


Fig. 12. The MC's learned trajectory when there is a mobile obstacle.

result shows that the MC's learned trajectory mainly passes through the regions with high distributed density of nodes, since the total number of charged nodes during the charging task mainly determines the MC's charging utility when each node maintains the similar initial battery level and the same average energy consumption rate.

Energy consumption of nodes: Fig. 10 is generated based on Scenario 5 where the initial battery level of each node is independently and identically sampled from the same Gaussian distribution. Also, the distributed density of nodes in different regions are the same. The nodes in the shape of a diamond have the higher average energy consumption rate than that in the shape of a pentagram. The result shows that the MC starts from the charging station and spends most time of the entire charging period in charging the nodes with high average energy consumption rates. Since the battery levels of nodes with higher average energy consumption rates drop faster over time, charging these nodes results in the higher charging utility achievable by the MC.

Multiple static obstacles and the mobility of obstacles: Fig. 11 and Fig. 12 are both generated based on Scenario 6. In Fig. 11, there are four static obstacles located in four different positions. The result shows that the proposed SAC-MSPI algorithm can find a safe and efficient moving trajectory for the MC to maximize the average effective charging rate while avoiding colliding with any obstacle. In Fig. 12, the mobile obstacle moves randomly along a certain trajectory, i.e., the black curve. The MC moves and charges nodes by following the learned trajectory, i.e., the red curve. We mark the real-time locations of both the MC and the mobile obstacle. The result shows that the MC can find an efficient moving trajectory without colliding with the mobile obstacle at any time instant during the entire charging period.

7 Conclusion

In this work, we maximized the MC's charging utility by jointly optimizing its moving trajectory and charging scheduling in a general and practical WRSN without accessibility to the exact system information. We designed an efficient model-free RL based algorithm called SAC-MSPI, which decouples the charging utility maximization and the collision risk minimization. Extensive evaluation results demonstrated that our designed algorithm outperforms existing main RL solutions and traditional baseline algorithms in terms of the charging utility and collision avoidance capability. We also validated the stability of SAC-MSPI in various experimental scenarios.

References

[1] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless Power Transfer via Strongly Coupled Magnetic Resonances," *Science*, vol. 317, no. 5834, pp. 83-86, Jul. 2007.

[2] L. Xie, Y. Shi, Y. T. Hou, and W. Lou, "Wireless Power Transfer and Applications to Sensor Networks," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 140-145, Aug. 2013.

[3] J. Huang, Y. Zhou, Z. Ning, and H. Gharavi, "Wireless Power Transfer and Energy Harvesting: Current Status and Future Prospects," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 163-169, Aug. 2019.

[4] N. Shinohara, "Trends in Wireless Power Transfer: WPT Technology for Energy Harvesting, Mllimeter-Wave/THz Rectennas, MIMO-WPT and Advances in Near-Field WPT Applications," *IEEE Microwave Magazine*, vol. 22, no. 1, pp. 46-59, Jan. 2021.

[5] C. Wang, J. Li, and Y. Yang, "Combining Solar Energy Harvesting with Wireless Charging for Hybrid Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 560-576, Mar. 2018.

[6] P. Zhou, C. Wang, and Y. Yang, "Self-sustainable Sensor Networks with Multi-source Energy Harvesting and Wireless Charging," in *Proc. of IEEE INFOCOM*, Paris, France, Apr. 2019.

[7] Y. Yang, and C. Wang, "Wireless Rechargeable Sensor Networks," *Springer*, 2015.

[8] Y. Sun, C. Lin, H. Dai, P. Wang, J. Ren, L. Wang, and G. Wu, "Recycling Wasted Energy for Mobile Charging," in *Proc. of IEEE ICNP*, Dallas, USA, Nov. 2021.

[9] C. Lin, S. Hao, H. Dai, W. Yang, L. Wang, G. Wu, and Q. Zhang, "Maximizing Charging Efficiency With Fresnel Zones," *IEEE Transactions on Mobile Computing*, Early Access, 2022.

[10] W. Yang, C. Lin, H. Dai, P. Wang, J. Ren, L. Wang, G. Wu, and Q. Zhang, "Robust Wireless Rechargeable Sensor Networks," *IEEE/ACM Transactions on Networking*, Early Access, 2022.

[11] Y. Sun, H. Dai, P. Wang, L. Wang, G. Wu, and Q. Zhang, "Trading off Charging and Sensing for Stochastic Events Monitoring in WRSNs," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 557-571, Apr. 2022.

[12] C. Lin, W. Yang, H. Dai, T. Li, Y. Wang, L. Wang, G. Wu, and Q. Zhang, "Near Optimal Charging Schedule for 3-D Wireless Rechargeable Sensor Networks," *IEEE Transactions on Mobile Computing*, Early Access, 2022.

[13] P. Yang, T. Wu, H. Dai, X. Rao, X. Wang, P. Wan, and X. He, "MORE: Multi-node Mobile Charging Scheduling for Deadline Constraints," *ACM Transactions on Sensor Networks*, vol. 17, no. 1, pp. 1-21, Article 7, Nov. 2020.

[14] T. Wu, P. Yang, H. Dai, C. Xiang, X. Rao, J. Huang, and T. Ma, "Joint Sensor Selection and Energy Allocation for Tasks-Driven Mobile Charging in Wireless Rechargeable Sensor Networks," IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11505-11523, Dec. 2020.

[15] Y. Shi, L. Xie, Y. T. Hou, and H. D. Sherali, "On Renewable Sensor Networks with Wireless Energy Transfer," in *Proc. of IEEE INFOCOM*, Shanghai, China, Apr. 2011.

[16] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the On-Demand Mobile Charging in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1861-1875, Sep. 2015.

[17] L. Chen, S. Lin, and H. Huang, "Charge Me If You Can: Charging Path Optimization and Scheduling in Mobile Networks," in *Proc. of ACM MobiHoc*, Paderborn, Germany, Jul. 2016.

[18] W. Liang, Z. Xu, W. Xu, J. Shi, G. Mao, and S. K. Das, "Approximation Algorithms for Charging Reward Maximization in Rechargeable Sensor Networks via a Mobile Charger," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 3161-3174, Oct. 2017.

[19] T. Wu, P. Yang, H. Dai, W. Xu, and M. Xu, "Collaborated Tasks-driven Mobile Charging and Scheduling: A Near Optimal Result," in *Proc. of IEEE INFOCOM*, Paris, France, Apr. 2019.

[20] H. Dai, Q. Ma, X. Wu, G. Chen, D. K. Y. Yau, S. Tang, X. Li, and C. Tian, "CHASE: Charging and Scheduling Scheme for Stochastic Event Capture in Wireless Rechargeable Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 44-59, Jan. 2020.

[21] Y. Ma, W. Liang, and W. Xu, "Charging Utility Maximization in Wireless Rechargeable Sensor Networks by Charging Multiple Sensors Simultaneously," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1591-1604, Aug. 2018.

[22] J. Liu, W. Xu, W. Liang, T. Liu, X. Peng, Z. Xu, Z. Li, and X. Jia, "Maximizing Sensor Lifetime via Multi-node Partial-Charging on Sensors," *IEEE Transactions on Mobile Computing*, Early Access, 2022.

[23] C. Lin, F. Gao, H. Dai, J. Ren, L. Wang, and G. Wu, "Maximizing Charging Utility with Obstacles through Fresnel Diffraction Model," in *Proc. of IEEE INFOCOM*, virtual conference, Jul. 2020.

[24] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Minimizing Charging Delay in Wireless Rechargeable Sensor Networks," in *Proc. of IEEE INFOCOM*, Turin, Italy, Apr. 2013.

[25] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Optimal Charging in Wireless Rechargeable Sensor Networks," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 278-291, Jan. 2016.

[26] C. Lin, Y. Zhou, F. Ma, J. Deng, L. Wang, and G. Wu, "Minimizing Charging Delay for Directional Charging inWireless Rechargeable Sensor Networks," in *Proc. of IEEE INFOCOM*, Paris, France, Apr. 2019.

[27] W. Xu, W. Liang, X. Jia, H. Kan, Y. Xu, and X. Zhang, "Minimizing the Maximum Charging Delay of Multiple Mobile Chargers Under the Multi-Node Energy Charging Scheme," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1846-1861, May. 2021.

[28] S. Wu, H. Dai, L. Liu, L. Xu, F. Xiao, and J. Xu, "Cooperative Scheduling for Directional Wireless Charging with Spatial Occupation," *IEEE Transactions on Mobile Computing*, Early Access, 2022.

[29] S. Zhang, Z. Qian, J. Wu, F. Kong, and S. Lu, "Optimizing Itinerary Selection and Charging Association for Mobile Chargers," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2833-2846, Oct. 2017.

[30] T. Liu, B. Wu, S. Zhang, J. Peng, and W. Xu, "An Effective Multi-node Charging Scheme for Wireless Rechargeable Sensor Networks," in *Proc. of IEEE INFOCOM*, Toronto, Canada, Jul. 2020.

[31] N. Wang, J. Wu, and H. Dai, "Bundle Charging: Wireless Charging Energy Minimization in Dense Wireless Sensor Networks," in *Proc. of IEEE ICDCS*, Dallas, USA, Jul. 2019.

[32] X. Wang, H. Dai, W. Wang, J. Zheng, N. Yu, G. Chen, W. Dou, and X. Wu, "Practical Heterogeneous Wireless Charger Placement with Obstacles," *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1910-1927, Aug. 2020.

[33] D. Mguni, U. Lslam, Y. Sun, X. Zhang, J. Jennings, A. Sootla, C. Yu, Z. Wang, J. Wang and Y. Yang, "DESTA: A Framework for Safe Reinforcement Learning with Markov Games of Intervention," https://arxiv.org/abs/2110.14468v3, Mar. 2023.

[34] S. He, J. Chen, F. Jiang, D. Yau, G. Xing, and Y. Sun, "Energy Provisioning in Wireless Rechargeable Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931-1942, Oct. 2013.

[35] S. Zhang, Z. Qian, F. Kong, J. Wu, and S. Lu, "P3: Joint Optimization of Charger Placement and Power Allocation for Wireless Power Transfer," in *Proc. of IEEE INFOCOM*, Kowloon, Hongkong, Apr. 2015.

[36] Z. Wang, L. Duan, and R. Zhang, "Adaptively Directional Wireless Power Transfer for Large-Scale Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1785-1800, May. 2016.

[37] H. Dai, X. Wang, A. X. Liu, H. Ma, G. Chen, and W. Dou, "Wireless Charger Placement for Directional Charging," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1865-1878, Aug. 2018.

[38] H. Dai, K. Sun, X. Liu, L. Zhang, J. Zheng, and G. Chen, "Charging Task Scheduling for Directional Wireless Charger Networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 11, pp. 3163-3180, Nov. 2021.

[39] N. Yu, H. Dai, G. Chen, X. Liu, B. Tian, and T. He, "Connectivity-Constrained Placement of Wireless Chargers," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 909-927, Mar. 2021.

[40] H. Dai, X. Wang, X. Lin, R. Gu, S. Shi, Y. Liu, W. Dou, and G. Chen, "Placing Wireless Chargers with Limited Mobility ," *IEEE Transactions on Mobile Computing*, Early Access, 2022.

[41] H. Dai, Y. Liu, G. Chen, X. Wu, T. He, A. X. Liu, and H. Ma, "Safe Charging for Wireless Power Transfer," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3531-3544, Dec. 2017.

[42] H. Dai, Y. Liu, N. Yu, C. Wu, G. Chen, T. He, and X. Liu, "Radiation Constrained Wireless Charger Placement," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 48-64, Feb, 2021.

[43] R. Jia, J. Wu, J. Lu, M. Li, F. Lin and Z. Zheng, "Energy Saving in Heterogeneous Wireless Rechargeable Sensor Networks," in *Proc. of IEEE INFOCOM*, Virtual Conference, May. 2022.

[44] W. Liang, W. Xu, X. Ren, X. Jia and X. Lin, "Maintaining Large-Scale Rechargeable Sensor Networks Perpetually via Multiple Mobile Charging Vehicles," *ACM Transactions on Sensor Networks*, vol. 12, no. 2, pp. 1-26, May. 2016.

[45] C. Lin, J. Zhou, C. Guo, H. Song, G. Wu and M. Obaidat, "TSCA: A Temporal-Spatial Real-Time Charging Scheduling Algorithm for On-Demand Architecture in Wireless Rechargeable Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 211-224, Jan. 2018.

[46] X. Cao, W. Xu, X. Liu, J. Peng and T. Liu, "A Deep Reinforcement Learning-based On-demand Charging Algorithm for Wireless Rechargeable Sensor Networks," *Ad Hoc Networks*, vol. 110, pp. 1-10, Jan. 2021.

[47] J. Chen, C. Yi, R. Wang, K. Zhu and J. Cai, "Learning Aided Joint Sensor Activation and Mobile Charging Vehicle Scheduling for Energy-Efficient WRSN-Based Industrial IoT," *IEEE Transactions on Vehicular Technology*, Early Access, 2023.

[48] T. Liu, B. Wu, W. Xu, X. Cao, J. Peng and H. Wu, "RLC: A Reinforcement Learning-Based Charging Algorithm for Mobile Devices," *ACM Transactions on Sensor Networks*, vol. 17, no. 4, pp. 1-23, Jul. 2021.

[49] Y. Liang, H. Wu and H. Wang, "ASM-PPO: Asynchronous and Scalable Multi-Agent PPO for Cooperative Charging," in *Proc. of AAMAS*, Online, May. 2022.

[50] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel and S. Levine, "Soft Actor-Critic Algorithms and Applications," https://arxiv.org/abs/1812.05905, Jan. 2019.

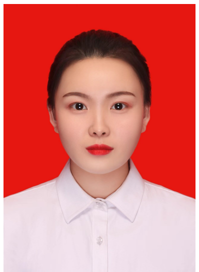
[51] A. Ray, J. Achiam and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," Open AI, https://openai.com/research/benchmarking-safe-exploration-in-deep-reinforcement-learning, Nov. 2019.

[52] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proc. of ICML*, Stockholm, Sweden, Jul. 2018.

[53] J. Schulman, S. Levine, P. Abbeel, M. Jordan and P. Moritz, "Trust Region Policy Optimization," in *Proc. of ICML*, Lile, France, Jul. 2015.

[54] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal Policy Optimization Algorithms," https://arxiv.org/abs/1707.06347, Aug. 2017.

[55] J. Achiam, D. Held, A. Tamar and P. Abbeel, "Constrained Policy Optimization," in *Proc. of ICML*, Sydney, Australia, Aug. 2017.



Xiuling Zhang received her M.S. degree in Electronics and Information Engineering from Zhejiang Normal University, China, in 2022. She is currently pursuing the Ph.D degree in College of System Engineering, National University of Defense Technology, Changsha, China. Her current research interests include smart IoT and reinforcement learning.

Riheng Jia (Member, IEEE, ACM) received his B.E. degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China, in 2012, and Ph.D. degree in Computer Science and Technology from Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently an Associate Professor with the School of Computer Science and Technology, Zhejiang Normal University, China. His current research interests include wireless networks, energy harvesting networks and smart IoT.

Quanjun Yin was born in Hunan, China, in 1978. He received the B.S., M.S., and Ph.D. degrees in simulation engineering from the College of System Engineering, National University of Defense Technology, Changsha, China, in 2008.

Zhonglong Zheng (Member, IEEE) received the B.E. degree from the China University of Petroleum, China, in 1999, and the Ph.D. degree from Shanghai Jiao Tong University, China, in 2005. He is currently a Full Professor with the School of Computer Science and Technology, Zhejiang Normal University, China. His research interests include machine learning, computer vision, and blockchain.

Minglu Li (Fellow, IEEE) received the Ph.D. degree in computer software from Shanghai Jiao Tong University in 1996. He is a Full Professor and the director of Artificial Intelligence Internet of Things (AIoT) Center at Zhejiang Normal University. He is also holding the director of Network Computing Center at Shanghai Jiao Tong University. He has published more than 400 papers in academic journals and international conferences. He was the chairman of Technical Committee on Services Computing (TCSVC) (2004-2016) and Technical Committee on Distributed Processing (TCD-P) (2005-2017), of IEEE Computer Society in Great China region. He served as a general co-chair of IEEE SCC, IEEE CCGrid, IEEE ICPADS, and IEEE IPDPS, and a vice chair of IEEE INFOCOM. He also served as a PC member of more than 50 international conferences including IEEE INFOCOM 2009-2016, IEEE CCGrid 2008, etc. His research interests include vehicular networks, big data, cloud computing, and wireless sensor networks. He is a fellow of IEEE.