

An efficient on-demand charging scheduling scheme for mobile charging vehicle

Ping Zhong¹  | Aikun Xu¹ | Jianliang Gao¹ | Yiming Zhang² | Yingwen Chen²

¹School of Computer Science and Engineering, Central South University, Changsha, China

²School of Computer, National University of Defense Technology, Changsha, China

Correspondence

Jianliang Gao, School of Computer Science and Engineering, Central South University, Changsha, Hunan 410083, China.

Email: gaojianliang@csu.edu.cn

Funding information

Natural Science Foundation of Hunan Province, Grant/Award Number: 2018JJ3692; National Natural Science Foundation of China, Grant/Award Number: 61402542; Fundamental Research Funds for Central Universities of the Central South University, Grant/Award Numbers: 2021zzts0735, 2021zzts0753

Summary

Shared transportation is a new way of traveling generated by the rapid development of the “Internet +” and the sharing economy. Due to the limited energy carried by shared e-bikes, the energy problem becomes the primary obstacle restricting its development. At present, one way to solve the energy problem is to use mobile charging vehicles (MCVs). However, how to schedule MCVs to achieve fast response, reduce energy consumption, and improve charging efficiency is crucial. In this paper, we propose a Charging Scheduling scheme based on Dynamic Characteristic of shared e-bike in Obstacle Space (CSDC-OS). Firstly, we design two MCV departure mechanisms to solve the problems of low utilization of MCVs and long waiting time for shared e-bikes. Then, we design the schedulability conditions based on the A-star algorithm and the dynamic change of e-bikes to ensure that MCVs can return to the service station in time. Finally, we propose a charging sequence algorithm based on linear skyline query and a selective insertion algorithm to improve the charging efficiency of MCVs. Simulation results show that the CSDC-OS can improve the response time, mobile consumption, and charging efficiency by more than 8% on average.

KEYWORDS

charging scheduling, mobile charging vehicle, shared e-bike, wireless rechargeable sensor network

1 | INTRODUCTION

The public bicycle transportation system is one of the fastest growing public transportation modes in the world, with an average annual increase of 37% since 2009.¹ China has the fastest growth rate, and the sales of electric bicycles (e-bikes) exceed other vehicles.^{2,3} E-bikes are more energy efficient and emit fewer greenhouse gas (GHG) per unit than other modes of transportation, such as cars, buses, and motorcycles.⁴ Shared bicycles, as a new mode of public transport that emerged in the past few years, are the most convenient and environmentally friendly short-haul public transport service. They have been deployed in major cities around the world thanks to their safety and cost-effectiveness.⁵

Following the sharing bicycles, shared electric bikes (shared e-bikes) become another popular option in the era of shared operations. Since shared e-bikes rely on batteries, the key factor restricting its normal operation is the remaining energy of the battery. How to recharge the battery is a challenge for the future operation of shared e-bike rental companies. Taking mobile charging vehicles (MCVs) to the service area where they can be charged when charging request is received is crucial, thus fundamentally solving the energy limitation problem of e-bikes. MCVs are vehicles that can

carry energy and be actively charged. With its special charging configuration, MCVs can perform fast charging and accomplish multiple charging tasks simultaneously. Although the price of MCVs is relatively high, they are still a popular choice due to their strong liquidity characteristics which can meet high charging demand in urban areas, and reduce the operation and maintenance cost to a certain extent for tram rental companies, hence saving manpower and material resources.

Researchers have done a lot of work on shared e-bikes. Ji et al.⁶ design a corresponding shared system based on the shared e-bike trial project at Knoxville, Tennessee (UTK) campus. Numerous scenarios were simulated for sensitivity analysis, and experiment results revealed the need to equip each e-bike with multiple replaceable batteries in areas of high pedestrian density in order to meet maximum user demand. Fyhri and Fearnley⁷ demonstrate that the use of shared e-bikes will increase the number of short trips made by the population and shared e-bikes have different benefits for different groups of people. Currently, research on shared e-bikes has focused on shared system design,^{8–13} shared e-bike structure design,^{14–18} and battery pack optimization.^{19–23} Research on the charging mode of the shared e-bike is still in its initial phase. Energy is the main problem that deters the development of shared e-bikes, leading to an increase in operating costs in the future.

Shared e-bikes are an infrastructure of the Internet of things. Central equipment, shared e-bikes, and people are connected via new efficient network connection methods, which finally realize the scientific and automatic connection of objects, people, and intelligent network.²⁴ Operators can centrally schedule and control e-bikes through central computers to achieve efficient post-management.

Wireless sensor networks (WSNs) are one of the prominent areas of the Internet of things. They are distributed wireless multi-hop networks composed of sensor nodes randomly distributed in the monitoring area. They have reliable transmission and intelligent processing and are widely used in various industries.²⁵ In traditional WSNs, sensor nodes mainly use batteries to maintain the normal operation. The limited battery capacity is the key factor that limits the entire lifetime of WSNs. Wireless rechargeable sensor networks (WRSNs) are a new type of network. They consist of a series of sensor nodes with energy harvesting (EH) combined with wireless power transmission (WPT)²⁶ to solve the node energy limitation problem in WSN.²⁷ The energy problem is also one of the main factors that limit the development of shared e-bikes. We regard the shared e-bikes as sensor nodes in WRSNs and conduct research based on the related work on WRSNs. The use of shared e-bikes has regional characteristics, and the locations constantly change. The urban structures of different cities differ, which results in complex changes in traffic flow within each city and the number of obstacles (such as buildings). The scheduling scheme for node energy supplementation in WRSNs therefore cannot be directly applied to the charging scheduling scheme for designing MCVs in the following operation of shared e-bikes. So optimizing the charging scheduling scheme of the MCVs in obstacle space brings extremely important value and practical significance to urban shared e-bike operation.

In order to solve the energy limitation problem of shared e-bikes in complex environments, we propose an efficient on-demand Charging Scheduling scheme based on Dynamic Characteristic of shared e-bike in Obstacle Space (CSDC-OS). The main contributions of our work are as follows:

- Discussing charging scheduling for the practical application scenarios of shared e-bikes. We design a charging scheduling optimization scheme under the scene of dynamic changes of shared e-bikes and varying obstacles.
- Proposing two dynamic starting mechanisms of MCVs. These mechanisms can fully adjust the charging request threshold and the task buffer pool size in combination with the number of charging requests in the network. Experimental results show that our methods can reduce charging delay, increase the MCV charging throughput, and improve resource utilization.
- Designing a charging path by combining the A-star algorithm with the dynamic changes of the e-bikes in obstacle space. The scheduling conditions proposed can ensure that MCVs return to the service station smoothly, thus increasing the feasibility of the charging scheduling scheme and reducing the risk of infeasibility.
- Designing a linear charging algorithm based on linear skyline query and a selective insertion algorithm to adjust the charging order of e-bikes in different situations. Simulation experiments show that these two algorithms can improve charging efficiency and reduce movement loss of MCVs.

The paper is organized as follows. Section 2 reviews some related work. Section 3 presents the problem description and system model. Section 4 introduces MCV starting mechanism. Sections 5 and 6 describe the charging sequence of e-bikes and the schedulability conditions of MCV charging. Section 7 presents the selective insertion algorithm. Section 8 evaluates the performance of the proposed solutions, and Section 9 concludes the paper.

2 | RELATED WORKS

The main problems faced by the shared e-bike are consistent with the main problems faced by the sensor nodes in the WRSN. The mobile energy source is used to supplement the energy of the sensor nodes in the WRSN. This method can fundamentally solve the energy problem in the WRSN, which is similar to the implementation process of using the MCV for the shared e-bikes charging. Designing a reasonable and efficient charging scheduling scheme for mobile energy sources is one of the basic research issues in WRSN. Charging scheduling refers to the reasonable design of the hardware configuration and software configuration in the WRSN to ensure that all nodes in the network can maintain normal operation during the target lifetime. Specifically, the hardware configuration is a reasonable design of various working parameters of the mobile charging source. The software configuration is to design an efficient mobile charging source charging scheme to maximize its utilization. Therefore, the goal of the charging scheduling problem is to achieve maximum network utility with limited resources. In practical applications, a reasonable and efficient charging scheduling scheme should be designed in combination with specific scenarios to maximize network utility.

2.1 | Maximize network utility–network lifetime

He et al.²⁸ laid the theoretical foundation for on-demand mobile charging (DMC) in WRSN. Specifically, they proposed a simple and effective “nearest-job-next with preemption” (NJNP) strategy. Using this strategy can analyze the problem of DMC. The simulation results show that NJNP can effectively extend the network lifetime. Lin et al.²⁹ considered the charging mode in the traditional WRSN to encounter bottlenecks in terms of charging throughput and lifetime. Aiming at this problem, a temporal and distasteful priority charging scheduling (TADP) algorithm considering the distance from the node to the MCV and the arrival time of the charging request was proposed. In addition, the two factors are gradually quantified. TADP formed a hybrid priority queue that directs the MCV to replenish energy for the node. The simulation proved that TADP can guarantee the success of higher priority task scheduling and improve the stability of the system. Liang et al.³⁰ considered deploying an MCV in the network and proposed an approximation algorithm to optimize the moving path to achieve the maximum energy, which extend the network lifetime. Lin et al.³¹ scheduled an MCV in a large-scale network. They proposed a candidate node and routing node scheduling algorithm, which not only solve the complex scheduling problem but also achieve higher charging throughput. Lin et al.³² considered using multiple MCVs to supplement energy for nodes and proposed a game cooperative scheduling strategy to determine the service nodes and service order, so that the throughput of MCV was maximized, thereby extending the network lifetime.

2.2 | Maximize network utility–MCV charging efficiency

Lin et al.³³ designed a collaborative charging scheme based on game theory for the problem that the lack of uniformity and dynamic charging strategy for collaborative charging schemes in WRSN. The charging scheme transformed the charging process into a collaborative process between MCV and studied the charging priorities, charging contributions and benefits between different MCVs to ensure that each MCV can achieve charging efficiency when complete charging tasks. Lin et al.³⁴ evaluated the lack of schedulability judgments in previous charging schemes, which would reduce the charging efficiency of MCV and lead to an increase in node mortality. In this problem, the author proposed a charging scheme based on optimal charging path planning and evaluates the schedulability of the charging task, so that the charging schedule can be predicted. Thus, it could increase the charging efficiency and the charging success rate of the MCV. Lin et al.³⁵ proposed the shortest path charging strategy for underwater WRSN to minimize the total moving distance of underwater robots. In addition, the author also proposed an emergency charging strategy to replenish energy for nodes that need to be recharged. The combination of the two strategies not only reduced the mobile energy consumption of the underwater robot but also improved the charging efficiency of the MCV.

2.3 | Minimize charging cost–deployment cost

Wang et al.³⁶ considered a hybrid framework network combining EH and WPT, where the cluster head node is powered by solar energy and the other nodes are powered by MCV. Using the approximate layout algorithm and

multiple time scheduling algorithms proposed in the paper can minimize the deployment cost. Liu et al.³⁷ comprehensively considering user needs, investment costs, soil location, emergency charging mileage limit, actual road conditions, service network reliability, and other factors, an MCV location optimization model with the dual goals of minimum investment cost and minimum user charging cost is established. Chen et al.³⁸ considered setting up charging MCV (responsible for charging nodes) and serving MCV (responsible for charging MCV) in large-scale networks and designed synergistic optimization of two MCV charging scheduling schemes to meet node charging requirements. Feasible algorithms have also been designed to reduce the total amount of MCV required. Liu et al.³⁹ introduced a new concept of “shutting” and proposed an optimization algorithm (push-shuttle-back [PSB]) in WRSN to minimize the number of required MCVs and get the best shutting distance (ignoring energy loss). Liang et al.⁴⁰ considered the MCV energy limitation and designed an on-demand scheduling strategy. This strategy was used to schedule MCV to charge nodes that need to replenish energy in a large-scale network, thereby minimizing the number of MCV deployments.

2.4 | Minimize charging cost–charging delay

Zhong et al.⁴¹ proposed a real-time on-demand charging scheduling scheme (RCSS) for the dynamic change characteristics of sensor nodes in the network. The MCV adjusted different nodes during the charging process and an adaptive charging threshold was used to reduce the charging delay. Fu et al.⁴² determined the staying range of the MCV by determining a minimum enclosed space containing all nodes and established a concentric circle centering on each node to determine the coincidence position of the concentric circles as the stopping position of the MCV. At the same time, the paper also considered that the location of the stay would cause the charging delay to increase. Finally, a scheme of combining adjacent residence positions is proposed to reduce the charging delay. Lin et al.⁴³ proposed a double warning thresholds with double preemption (DWDP) charging strategy for time and space constraints in the charging process. For the remaining energy of different sensor nodes, DWDP adjusted the node charging priority by adjusting the warning threshold to ensure high throughput and low charging delay.

The above researches usually ignore the impact of the MCV starting mechanism on the entire charging scheduling scheme. Most of them are based on the fixed task buffer pool or fixed energy threshold to design MCV charging scheduling scheme. In the actual scenario, due to different charging requirements in different cities, the MCV starting mechanism will have an impact on the overall charging performance. Therefore, how to dynamically adjust the task buffer pool size and charging energy threshold according to environmental changes is a problem that needs to be discussed in this paper. In addition, existing researches usually discuss MCV charging scheduling in a single-factor or ideal environment, which also leads to existing researches not applicable to complex and variable shared e-bike operating environments. Therefore, how to discuss a feasible charging scheme in a complex environment is a key step to solve the energy problem of the shared e-bike.

3 | PROBLEM DESCRIPTION AND SYSTEM MODEL

3.1 | Problem description

In practical scenarios, shared e-bikes are in the process of continuous dynamic change, and their location has the characteristics of time-varying because of the frequent use. In addition, there are a lot of obstacles (such as buildings) in the urban area. Thus, the research on charging scheduling of MCV for static sensor nodes in ideal environment is no longer applicable. We propose a charging scheduling scheme based on dynamic characteristics of shared electric bicycle in obstacle space, which mainly solves problems as follows:

3.1.1 | MCV departure mechanism based on dynamic change of shared e-bike

There are two main MCV departure mechanisms in the existing researches. One is that the MCV receives the charging request and then starts to serve the node that needs to be charged. The second is that the MCV has a fixed task buffer pool. When the task buffer pool is full, it starts to perform charging tasks in the task buffer pool. These two departure mechanisms have their inherent shortcomings. The first departure mechanism causes the MCV to start frequently and

waste resources when the charging load is large. The second departure mechanism causes the node charged to wait too long when the charging load is small. Designing a reasonable and efficient departure mechanism can improve MCV utilization and reduce path loss. Departure from the shared e-bike and MCV, we design two adaptive MCV departure mechanisms for different needs, which provide an important basis for optimizing the charging scheduling scheme.

3.1.2 | Reasonable and efficient shared e-bike charging order determination

We consider that the MCV only serves one e-bike at a time. The current charging order of the e-bike will affect whether the subsequent e-bike can be charged before the energy exhausted. A reasonable and efficient charging sequence of shared e-bike is very important to improve the charging throughput of MCV. The main factors affecting the charging order we consider are the frequency of use in the past, the distance from MCV to e-bike, and the remaining energy. How to design a reasonable charging sequence based on these influencing factors is a key step in the overall charging scheduling scheme.

3.1.3 | MCV charging path selection in obstacle space

In the actual environment, a large number of obstacles cause the shared e-bike to be not directly reachable and is also in a process of constant dynamic changes. Considering that the main energy consumption of the MCV is used for path movement, selecting a feasible path for the MCV is very important for reducing the MCV path loss and improving the MCV charging efficiency. In addition, the number of shared e-bikes that the MCV can serve is limited due to the limited energy of the MCV. It is necessary to design a suitable charging path for the MCV, which is to ensure that the MCV returns to the service station smoothly for energy replenishment.

3.2 | System model

It is assumed that N ($N > 1$) shared e-bikes and S square obstacles with side length r are distributed in a two-dimensional rectangular area. Each shared e-bike is represented by N_i ($i = 1, 2, \dots, N$), and the shared e-bike collection is $N = \{N_1, N_2, \dots, N_N\}$. As shown in Figure 1, each shared e-bike is charged one-to-one by the MCV. In order to simulate the continuous motion process of the e-bike in the real scene, it is assumed that the e-bike is moving at a uniform speed of v_{SEB} . The MCV moves from the service station SS at the speed of v_M to the e-bike that needs to be charged. The service station SS is located near the first served e-bike, ignoring the distance between the service station and the first served e-bike. The battery capacity of the e-bike is E_{SEB} , and the MCV selects the next e-bike to be serviced after charging the serviced e-bike to E_{SEB} . There may be a special case, the MCV receives a charging request from a shared e-bike

TABLE 1 List of notations

Notation	Definition
N	Set of shared e-bikes in the two-dimensional rectangular area
E_{SEB}	The battery capacity of the e-bike
E_T	The threshold for shared e-bike ($E_T > E_L$)
E_L	The threshold for shared e-bike
v_M	The moving speed of the MCV
v_{SEB}	The moving speed of the shared e-bike
q_M	The mobile energy consumption of the MCV
q_c	The charging rate
B_{minthr}	The minimum task buffer pool size of MCV
η	The energy receiving efficiency
w	The weighting factor

(candidate charging e-bike) along the way while MCV moves to the next charging e-bike. The angle between the two sides is γ , where one side is the connection between the MCV and the candidate charging e-bike and another side is the connection between the next charging e-bike and the candidate charging e-bike. In this paper, we consider that when the cosine of the offset angle threshold γ is approximately negative 1, which means the lower limit of γ is 90° and the upper limit of γ is 270° . In this case, the candidate charging e-bike can be inserted into the MCV service queue. A detailed explanation is in Section 7. Each time the MCV completes the charging task or the energy is insufficient, it returns to the service station SS to replenish its own energy, supplements to the maximum battery capacity and waits for the next charging schedule. The related notation definitions are shown in Table 1.

4 | MCV DEPARTURE MECHANISM BASED ON DYNAMIC CHANGE OF E-BIKE

In the existing related research, the departure service mechanism for MCV usually considers two options. One is to start the service for the e-bike in the task buffer pool when the charging request received by the task buffer pool of the MCV is full. In this solution, the task buffer pool cannot be full when the workload is small, which may cause a large charging delay of some e-bike that need to be charged. The second is that the MCV immediately starts to service the e-bike after receiving the charging request and dynamically selects the next charging e-bike during the charging process. This kind of scheme will lead to a more frequent charging process when the workload is low, which results in wasted MCV resources and low charging efficiency. In view of the problems of these two schemes, we propose the MCV departure mechanism based on the adaptive task buffer pool and the MCV departure mechanism based on a double threshold.

4.1 | MCV departure mechanism based on adaptive task buffer pool

To illustrate the charging process based on the adaptive task buffer pool, we take Figure 2 as an example. In the initial situation, the MCV task buffer pool size is the minimum task buffer pool size B_{minthr} . MCV selects the e-bike N_i to be served when the task buffer pool size is full and sends CONFIRM message to N_i . At this time, N_i stops using and MCV serves it. Whenever the MCV completes the charging service for the e-bike N_i , the task buffer pool size is updated automatically according to the number of received charging requests. Then, the newly received charging request is added according to the updated task buffer pool size. The MCV task buffer pool update process occurs during the entire MCV execution of the charging task until the MCV returns to the service station due to insufficient energy or completion of the charging task.

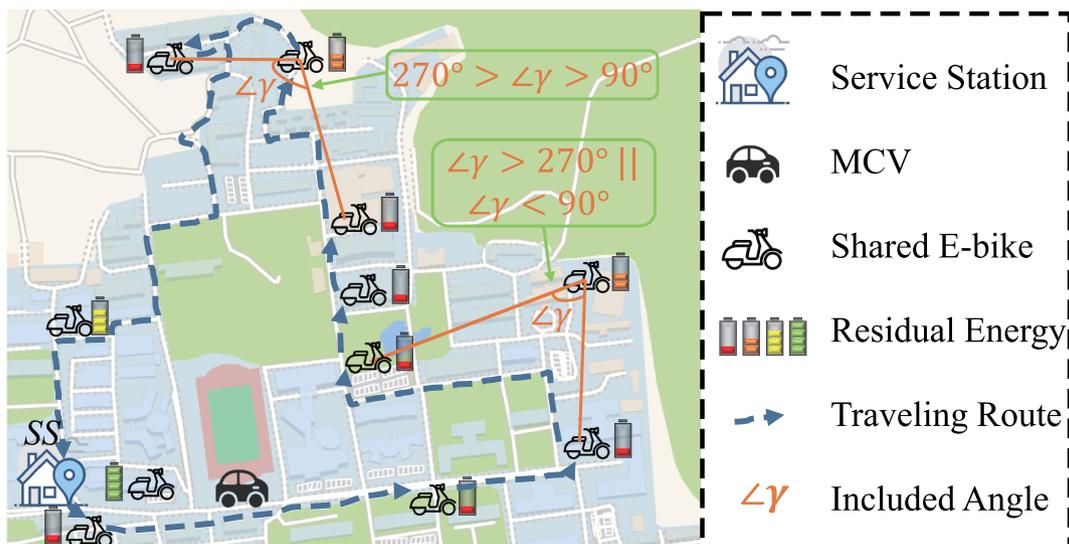


FIGURE 1 The scene of shared e-bike model under MCV charging

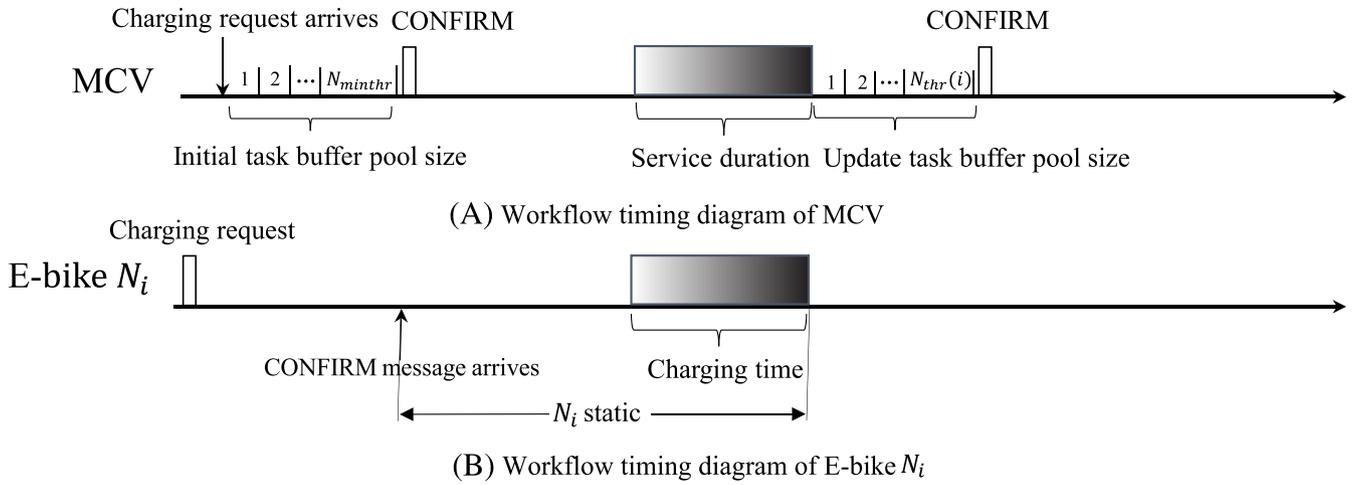


FIGURE 2 Workflow timing diagram based on adaptive task buffer pool MCV departure mechanism

We assume that the MCV has a maximum task buffer pool size B_{maxthr} and a minimum task buffer pool size B_{minthr} . B_{maxthr} means that the maximum number of e-bikes can be served. Since the energy consumed by the MCV during the movement is much larger than the energy used for charging. As shown in Equation (1), B_{maxthr} is equivalent to the number of e-bikes that can be serviced with the remaining energy after the total energy carried by the MCV minus the energy consumed by the MCV moving the farthest distance,

$$B_{maxthr} = \left\lfloor \frac{E_M - \left(\frac{D_{max}}{v_M}\right) \times q_M}{E_{SEB}} \right\rfloor, \quad (1)$$

where E_M is the residual energy of the MCV and the initial value of E_M is 50,000 J. D_{max} is the farthest moving distance of the MCV, v_M is the moving speed of the MCV, q_M is the mobile energy consumption of the MCV, and E_{SEB} is the total capacity of the shared e-bike.

B_{minthr} is a fixed value and means the minimum number of e-bikes can be served and $B_{minthr} = 4$. In order to make the MCV task buffer pool size adaptively adjusted according to the number of charging requests in the current network, as shown in Equation (2), we set a threshold ψ_i according to the number of charging requests,

$$\psi_i = \frac{n_i}{|N|}, \quad (2)$$

where n_i is the number of charging requests that are not yet served by MCV. As shown in Equation (3), n_i includes the charging task n_{old} that has been added to the task buffer pool and the number of newly received charging requests n_{new} . $|N|$ is the total number of shared e-bikes. After the MCV finishes the service for e-bike N_i , its adaptive task buffer pool size $B_{thr}(i)$ is adjusted as shown in Equation (4):

$$n_i = n_{old} + n_{new}, \quad (3)$$

$$B_{thr}(i) = \lceil \psi_i (B_{maxthr} - B_{minthr}) \rceil + B_{minthr}. \quad (4)$$

In short, the MCV departure mechanism based on adaptive task buffer pool has two obvious advantages. On the one hand, when there is less load on the charging task in the network, the method of adaptively adjusting the size of the task buffer pool according to the number of charging requests can reduce the waiting time for the shared e-bike to wait for charging time by reducing the size of the task buffer pool. On the other hand, when there are many charging tasks, the method can increase the size of the task buffer pool according to the number of charging tasks, thereby increasing the throughput of the MCV.

4.2 | MCV departure mechanism based on dual threshold

To solve the problem of MCV wasting resources due to frequent departures, we set two energy thresholds for shared e-bike, namely, E_L and E_T , and $E_L < E_T$. The charging process based on dual threshold is explained by Figure 3. When the remaining energy of the e-bike N_i is lower than E_T , the e-bike N_i sends a charging request to the MCV. Regardless of the MCV task buffer pool size, the received charging request is placed into the MCV task buffer pool. When the remaining energy of the e-bike N_i is lower than E_L , N_i sends an ALERT message to the MCV, indicating that its energy has reached the warning value. The MCV starts a round of charging after receiving the first ALERT message. The MCV will determine the service order according to the charging order determination algorithm based on the skyline query described below. After determining that the e-bike to be served first is e-bike N_j , the CONFIRM message is sent to inform N_j that it will be served, and N_j stops working and waits for the MCV to charge it. After completing the charging service of N_j , the MCV determines that the next e-bike to be served is N_i and sends CONFIRM message to N_i . Then, MCV moves to N_i for service. This process is repeated until the MCV completes all charging requests or returns to the service station due to insufficient energy.

Considering that E_L is more important than E_T , we set E_T as a fixed value and the value of E_L is solved by numerical simulation because E_L determines when the MCV starts to perform the charging task. Specifically, the value of E_L is dynamically calculated by using the average energy consumption rate (AECR) and the average failure probability (AFP) per unit time. The numerical simulation is used to solve the change of the AECR and AFP values in the process of adjusting the value of E_L and get the best value of E_L . As shown in Equations (5) and (6), we apply curve fitting to the simulation results in order to derive two approximate linear relations of AECR and AFP in terms of E_L :

$$AECR(E_L) \approx A_1 E_L + B_1, \quad (5)$$

$$AFP(E_L) \approx A_2 E_L + B_2, \quad (6)$$

where A_1 , A_2 , B_1 , and B_2 are parameter values. As shown in Equation (7), the multi-objective optimization method is used to optimize the value of E_L .⁴⁴ We aim to minimize the values of AECR and AFP, introduce the weighting factor w , and use the square sum equation in the least squares optimization method to optimize the solution,

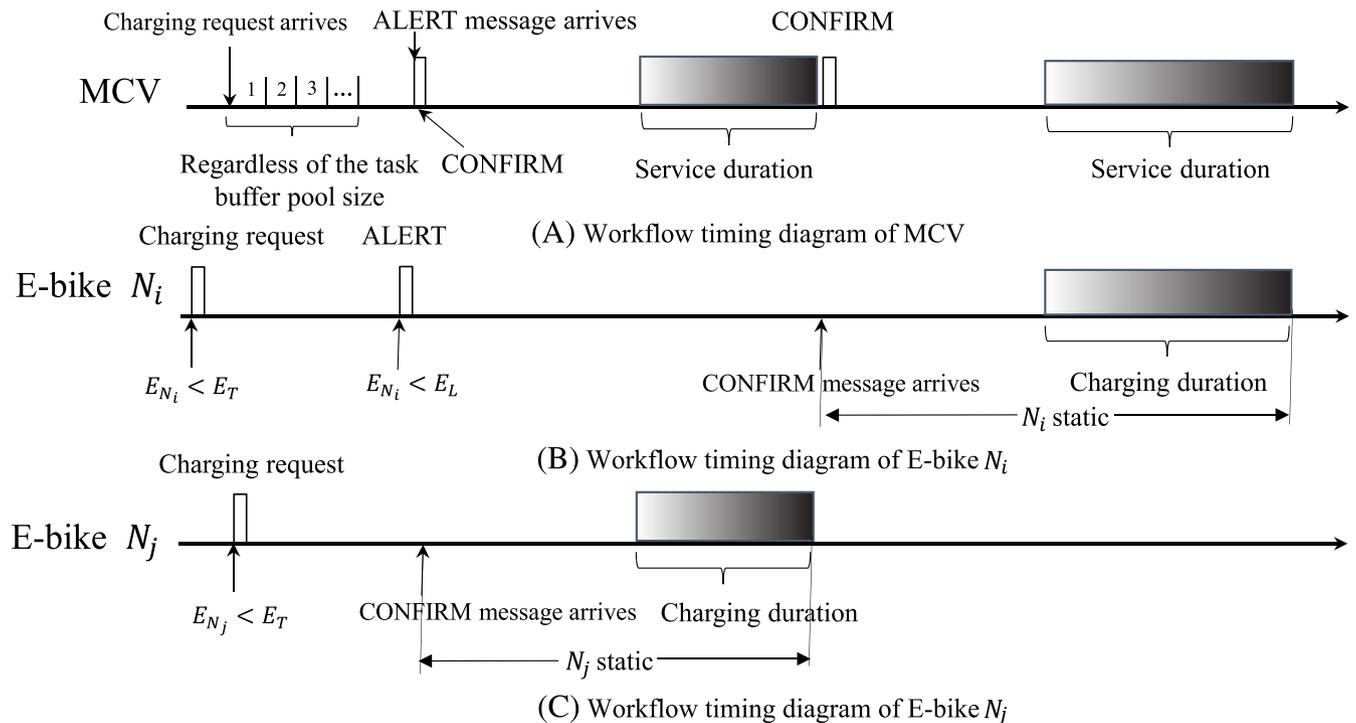


FIGURE 3 Workflow timing diagram based on dual-threshold MCV starting mechanism

$$f(w) = (AECR)^2 + w(AFP)^2. \quad (7)$$

The optimal E_L is able to minimize the value of the objective function in Equation (7). As shown in Equation (8), the optimal E_L can be obtained by bringing Equations (5) and (6) into Equation (7):

$$E_L(w) = -\frac{A_1B_1 + wA_2B_2}{(A_1)^2 + w(A_2)^2}. \quad (8)$$

5 | THE CHARGING SEQUENCE OF E-BIKE BASED ON LINEAR SKYLINE QUERY

The core idea of linear skyline query method is to consider a variety of factors affecting the charging order of e-bike and try to find out the charging sequence which conforms to the definition of linear skyline query from all feasible service electric bicycles.

The definition of the linear skyline query is derived from the definition of the skyline query. In the definition of the skyline query, the charging order **A** dominates the charging order **B** if and only if the charging order **A** is superior to the charging order **B** in one or several metrics and is not weaker than the charging order in other metrics.⁴⁵ Our goal is to find the e-bike that meets the corresponding conditions from all feasible charging sequences. These e-bikes constitute the target set. The e-bike in the target set is not inferior to the metrics of the e-bikes in the nontarget set. In order to introduce the definition of skyline query more clearly, we introduce two main metrics as an example. We consider the main factors that affect the charging order are the remaining energy $E_{Residual}$ of the e-bike and the distance d_{MCV} to the MCV. The e-bike with the least remaining energy and the closest to the MCV will be first served. We use the skyline query to reduce the number of target sets, and the elements in the target set are called interest data points. As shown in Figure 4, the target set is $\{a, i, k\}$, where node a has the least remaining energy, and the distance from node k to MCV is the closest. Node i has neither the shortest distance nor the least remaining energy, but the residual energy of node i is lower than k and the distance to the MCV is close to a . Thus, node i is not dominated by node a and node k . The remaining other data points will be dominated by the target set $\{a, i, k\}$, which means that the data points, whether the remaining energy or the distance, are greater than one or more elements in the interest data points.

The linear skyline query is to further narrow the scope of the target set, which means that under all linear combinations, the metrics of the e-bike in the target set must be optimal. Formally, there are \mathbf{R} possible charging orders in the target set. The \mathbf{R} possible charging orders can be represented by \mathbf{R} cost functions. In the cost function, c_1, c_2, \dots, c_t are the t -tuples of the metric, and t is the metric that affects the charging order. In this paper, we consider the frequency of use, the remaining energy, and the distance to the MCV as the main indicators affecting the charging order of the e-bike, that is, $t = 3$. For order **A** and order **B** in a feasible charging sequence **P**, the cost use vector of order **A** can be represented by $[c_1(A), c_2(A), c_3(A)]^T$ and the cost use vector of order **B** can be represented by $[c_1(B), c_2(B), c_3(B)]^T$.

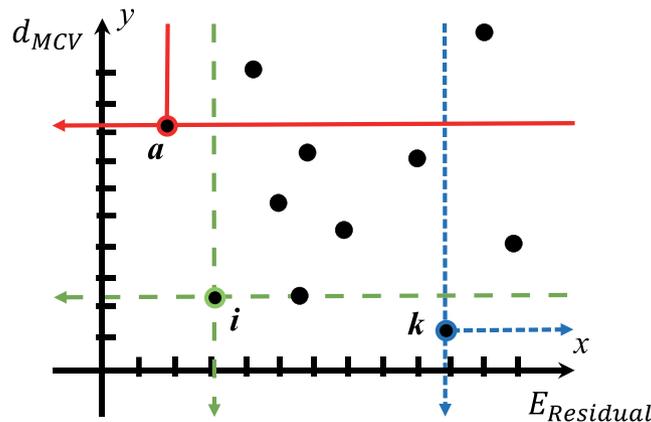


FIGURE 4 Skyline query example illustration

The charging order **A** dominates the charging order **B** if and only if $\exists 1 \leq i \leq 3 : c_i(A) < c_i(B)$ and $\nexists 1 \leq i \leq 3 : c_i(A) > c_i(B)$. The result for the skyline query is a subset of all the e-bike sets **P** that needs to be charged, and the subset is $P_C = \{x \in P \mid \nexists y \in P : x < y\}$. The shared e-bike charging order determination algorithm occurs when the MCV completes the charging service for the current e-bike and the e-bikes have updated the data (remaining energy, distance to the MCV, and the frequency of use).

6 | MCV CHARGING SCHEDULABILITY CONDITIONS

Whether the MCV starting mechanism is based on the adaptive task buffer pool or the dual threshold, the MCV immediately starts to service the e-bike when the conditions of the starting mechanism are met. The main energy consumption of MCV is mobile energy consumption and charging energy consumption, in which mobile energy consumption is much larger than charging energy consumption. After the MCV performs the charging task, it needs to ensure that the MCV can smoothly return to the service station *SS* for its own energy supplement. Therefore, when the e-bike N_i is determined as the next service e-bike, it is necessary to determine whether the remaining energy of the MCV after the service of the e-bike N_i can ensure that the MCV returns to the service station *SS* smoothly.

In the actual urban scene, the e-bikes of adjacent charging order are not directly reachable due to the presence of obstacles. When MCV chooses a moving path, it faces a large search space. In this paper, we use A-star algorithm to solve this path search problem, which can reduce the complexity of search space and find the shortest distance between two points in the obstacle space. A-star algorithm is very suitable for large-scale urban traffic path search, which combines the advantages of Dijkstra algorithm and BFS search algorithm to improve the flexibility of path search. When the A-star algorithm is applied to the path search, the entire search space can be divided by a square, a triangle, and a hexagon. For convenience, we consider gridding the entire search space, using a single square grid as the basic unit of pathfinding. By setting the cost function to describe the distance or difficulty of the moving path, the obstacle can be regarded as a basic unit of several pathfinding that cannot pass. The goal of the A-star algorithm is to find the least costly path in the obstacle space. In this paper, we consider designing the MCV moving path with the diagonal distance as a cost function.

The shared e-bike can still be used after sending the charging request. The charging request is sent only to inform the service station that there is not much energy left. Before the MCV determines its service, its position is still in a dynamic state. This feature makes it impossible to use the A-star algorithm directly in the obstacle space because the precondition for the A-star algorithm to find the best path between two points in the obstacle space is the known starting point and the ending point. However, for the dynamically changing e-bike, it is impossible to know exactly where the destination is. Therefore, we introduce the concept of single-serving time for e-bike, which is defined as the sum of the time when the MCV moves to the next charging e-bike and the charging duration for the e-bike. Through the initial position, moving speed, rotation direction, and single running time of the e-bike, the position of the e-bike and the distance to the MCV can be obtained.

Since the MCV has limited energy, it is necessary to ensure that the MCV can return to the service station for energy supplementation before the energy is exhausted. Therefore, it is necessary to determine whether the MCV can smoothly return to the service station after determining the next service e-bike and the best path to move. Assuming that there are n charging tasks in the task buffer pool, the MCV charging period T is as shown in Equation (9),

$$T = t_n + \tau_n + \frac{d_{n,1}}{v_M} + T_{stay}, \quad (9)$$

where T_{stay} refers to the length of time that the MCV returns to the service station for energy replenishment and waits for the next scheduling. v_M is the moving rate of the MCV, t_n is the time when the MCV reaches the n th e-bike, and τ_n is the service time for the n th e-bike. $d_{n,1}$ is the path length for MCV from the last served e-bike to the service station. The time when the MCV reaches the e-bike N_i is shown in Equation (10),

$$t_i = t_{i-1} + \tau_{i-1} + \frac{d_{i-1,i}}{v_M} \quad 2 \leq i \leq N. \quad (10)$$

Use Equations (9) and (10) to generalize to the entire charging cycle T as shown in Equation (11),

$$T = \frac{\sum_{i=1}^n d_{i,i+1} + d_{n,1}}{v_M} + \sum_{i=1}^n \tau_i + T_{stay}. \quad (11)$$

In order to ensure the normal operation of the e-bike, it should be ensured that the remaining energy of the e-bike will not become zero before being charged. We consider from the worst case; that is, the e-bike N_i to be charged is always used and the remaining energy continues to drop. As shown in Equation (12), as long as the remaining energy does not become zero, in this case, the normal operation of e-bike N_i can be ensured,

$$E_T - p_i \times t_i > 0, \quad (12)$$

where E_T is the energy threshold for the charging request and p_i is the mobile energy consumption of the e-bike. The e-bike does not consume energy during the charging process. The energy change of the e-bike during the charging process is as shown in Equation (13),

$$E_{SEB} = E_T - t_i \times p_i + \tau_i \times q_c \times \eta, \quad (13)$$

where E_{SEB} is the total capacity of the e-bike, q_c is the charging rate, and η is the energy receiving efficiency. In order to ensure the fairness of charging, the shared e-bike is only serviced once during each round of charging. As shown in Equation (14), it is necessary to ensure that the remaining energy of the e-bike will not reach E_T until the next round of charging begins,

$$E_{SEB} - p_i \times (T - t_i - \tau_i) > E_T. \quad (14)$$

Substituting $E_{SEB} = E_T - t_i \times p_i + \tau_i \times q_c \times \eta$ into Equation (14) can obtain Equation (15),

$$\tau_i \times q_c \times \eta > p_i \times (T - \tau_i). \quad (15)$$

Equation (16) can be obtained by generalizing Equation (13) to the entire charge cycle,

$$\sum_{i=1}^n \tau_i \times q_c \times \eta > p_{sum} \times \sum_{i=1}^n (T - \tau_i), \quad (16)$$

where p_{sum} is the total energy consumption rate. The entire charging cycle includes the MCV travel time and charging time. As shown in Equation (17), the time that the MCV is used for movement is greater than the duration of the charging service,

$$\sum_{i=1}^n \tau_i < \sum_{i=1}^n (T - \tau_i). \quad (17)$$

Equation (18) can be obtained by combining Equations (16) and (17), which can be ensured that the e-bike will not deplete energy before being serviced,

$$q_c \times \eta > p_{sum}. \quad (18)$$

In addition, it is necessary to ensure that the MCV has sufficient energy to return to the service station. As shown in Equation (19), the total energy of MCV cannot be less than the sum of the charging energy and the path moving energy,

$$E_M \geq p_{sum} \times \frac{D \times q_c}{v_M \times (q_c \times \eta - p_{sum})} + \frac{D}{v_M} \times q_M, \quad (19)$$

where D is the length of the entire loop and q_M is the mobile energy consumption of the MCV.

In summary, the MCV path schedulability condition needs to satisfy Equations (18) and (19). If the path schedulability condition is met, the MCV can determine its service.

7 | SELECTIVE INSERTION ALGORITHM FOR SHARED E-BIKE

Due to the regional differences in the frequency of using shared e-bike, the high-frequency use of e-bike appears in some areas, such as schools and office areas. There may be a special case in the real scene. The MCV receives a charging request along the way, while MCV moves to the next charging e-bike. The location where the charging request is issued is close to the target service e-bike. The insertion of such e-bike that needs to be charged into the charging sequence has little effect on the target service e-bike. For this special case, we consider adjusting the charging ordering according to the geographical location of the shared e-bike.

Specifically, as shown in Figure 5A, when the MCV selects the e-bike N_j as the target service e-bike, it receives the ALERT message from the e-bike N_i on the way to N_j , which indicates that N_i also needs to charge urgently. During the MCV operation, the energy consumption of the MCV during operation is mainly concentrated on the mobile energy consumption and the charging energy consumption. In addition, the charging energy consumption is much smaller than the mobile energy consumption. The feasible path from MCV to N_i is mainly thought when considering whether N_i can be served before N_j . The distance from the position of MCV (*i.e.*, D_{MCV}) to the target service e-bike N_j can be obtained by Equation (20). As shown in Figure 5B, when the angle between the two sides is within a certain range, the cosine value is approximately negative 1 based on the cosine theorem, $\cos x \in [-1, 1]$. In this case, Equation (20) can be transformed into Equation (21) and then reduced to Equation (22) according to the square sum formula. Therefore, we set the offset angle threshold γ . If $\angle N_i D_{MCV} N_j$ satisfies the threshold γ , the MCV will serve the e-bike N_i first and then serve the original target charging e-bike N_j . The energy consumption of MCV is similar to the energy consumed directly by the target e-bike N_j service. The offset angle threshold γ is the same as the cosine of the complementary angle, that is, $\cos \gamma = \cos(2\pi - \gamma)$. In this paper, we consider that when the cosine of the offset angle threshold γ is approximately negative 1, which means the lower limit of γ is 90° and the upper limit of γ is 270° . In this case, the e-bike N_i can be inserted into the MCV service queue.

$$d_{D_{MCV}N_j} = \sqrt{d_{D_{MCV}N_i}^2 + d_{N_iN_j}^2 - 2d_{D_{MCV}N_i}d_{N_iN_j}\cos\gamma}, \quad (20)$$

$$d_{D_{MCV}N_j} \approx \sqrt{d_{D_{MCV}N_i}^2 + d_{N_iN_j}^2 + 2d_{D_{MCV}N_i}d_{N_iN_j}}, \quad (21)$$

$$d_{D_{MCV}N_j} = d_{D_{MCV}N_i} + d_{N_iN_j}, \quad (22)$$

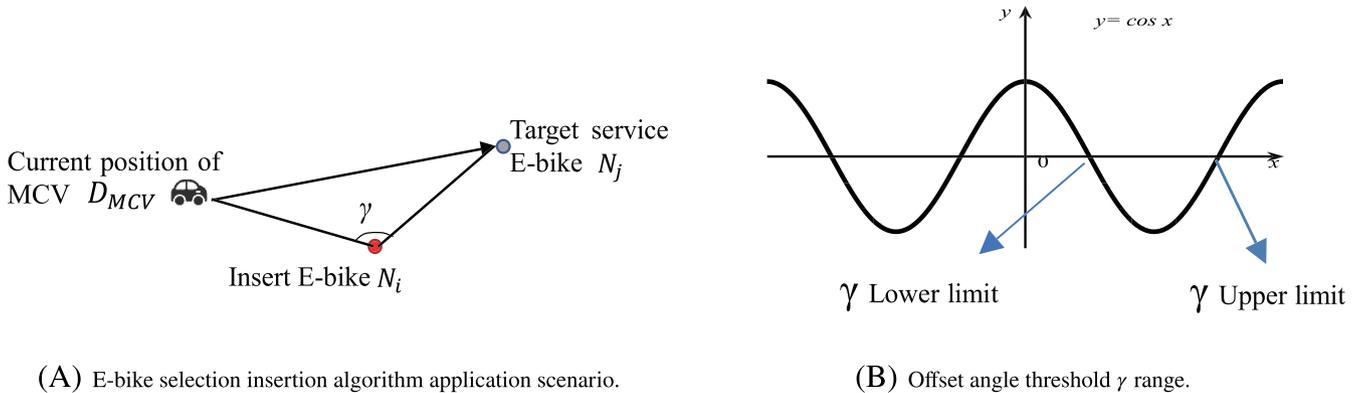


FIGURE 5 E-bike selection insertion algorithm position diagram

where $d_{D_{MCV}N_j}$ is the distance from the MCV location to the target service e-bike N_j , $d_{D_{MCV}N_i}$ is the distance from the MCV location to the insertion service e-bike N_i , and $d_{N_iN_j}$ is the distance from the insertion service e-bike N_i to the target service e-bike N_j .

8 | EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

We use MATLAB and Python to implement the charging scheduling scheme in this paper. It is considered to randomly generate several nodes and 20 square obstacles with a side length of 10 m in a 500×500 m square network area to describe shared e-bikes and buildings in the actual scene. Each shared e-bike moves in the network at the same rate. In the initial case, all e-bikes are rotated counterclockwise by an integral multiple of 30° in the positive direction of the x axis as the moving direction. During the movement, if the e-bike encounters an obstacle, it will avoid the obstacle by superimposing the 30° direction angle counterclockwise. In order to simulate the different frequencies used between different e-bikes, we set the mobile energy consumption of e-bike to a random value between 0.25 and 0.8 J/s. Once the MCV determines the next e-bike to be serviced, it will send CONFIRM message to the e-bike. At the same time, the e-bike stops moving and waits for the MCV service. Otherwise, the e-bike is being used by the user or waiting to be used. Table 2 shows the experimental specific parameter settings.

8.1 | E_L optimization based on dual-threshold MCV starting mechanism

The warning threshold E_L of the e-bike can be obtained by numerical simulation. As shown in Figure 6, we set the value of E_T to a fixed value of 225 J, and E_L varies between $[0, 220]$. AECR increases with the increase of E_L because MCV waits for a longer time to receive more charging requests when $E_L \ll E_T$. The MCV starts to perform the charging task only when the remaining energy of the motorcycle reaches the E_L . For more charging tasks, the resource utilization of the MCV and the AECR of the e-bike are improved. In addition, the MCV waits for more charging requests, which causes the charging waiting time to be longer and increases the charging delay. Therefore, the AFP decreases as the E_L increases. In summary, as the value of E_L gradually increases and approaches the value of E_T , the number of charging requests collected by the MCV gradually decreases and the MCV waits for less time to start immediately after receiving the first charging request, which can reduce the value of AFP. Moreover, MCV has fewer e-bikes to serve after departure, which means more e-bikes work normally in the network. Thus, the value of AECR increases.

Table 3 lists the values of the best E_L and its corresponding w obtained under the number of e-bikes. The value of E_L decreases as the number of e-bikes increases because the MCV needs to collect more charging requests from the e-bike to ensure that the optimal objective function value is achieved when the number of e-bikes increases. Thus, it needs to reduce the value of E_L to ensure that the number of charging requests is sufficient.

TABLE 2 Parameter settings

Parameter symbol	Value
N	40–120
E_{SEB}	500 J
E_T	225 J
v_M	5 m/s
v_{SEB}	3 m/s
q_M	30 J/s
q_c	5 J/s
B_{minthr}	4
η	0.5

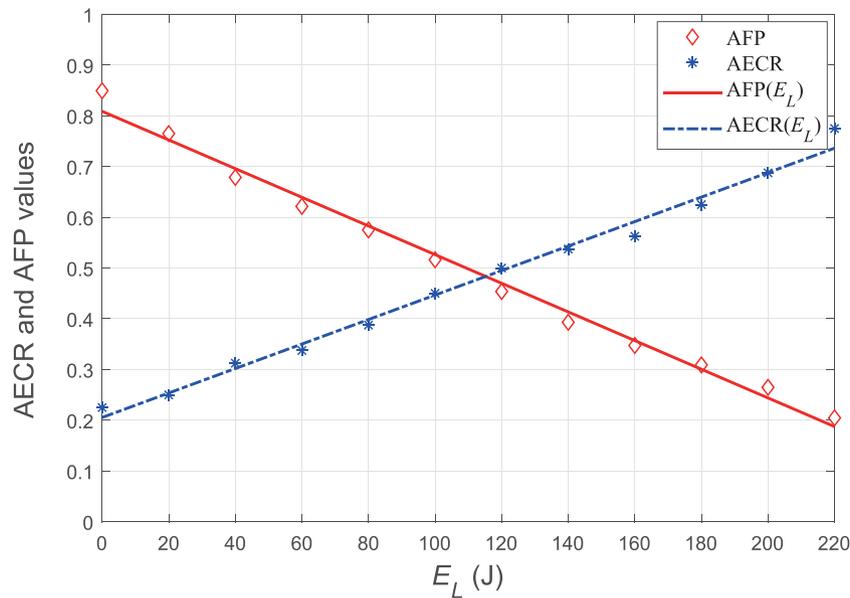


FIGURE 6 The effect of E_L change value on AECR and AFP

TABLE 3 The best E_L value for different numbers of e-bikes

Number of e-bikes	w	E_L
40	73.6	101.87 J
60	86.1	96.18 J
80	70.1	78.03 J
100	68	63.54 J
120	74	65.21 J

8.2 | The impact of the shared e-bike number N

In this paper, the charging scheduling scheme under the adaptive task buffer pool of the MCV is called CSDC-OS1 and the charging scheduling scheme under the dual threshold is called CSDC-OS2. In order to study the influence of different e-bike numbers on the charging performance of the MCV under the CSDC-OS scheme, we compare the MCV moving duration and charging duration under different starting mechanisms. As shown in Figure 7, the moving time of the MCV increases as the number of e-bikes increases because an increase in the number of e-bikes will increase the MCV workload, thereby increasing the path movement time. In addition, it can be seen that the movement time of the MCV under the CSDC-OS1 is much larger than that of the CSDC-OS2. The MCV serves for e-bikes in the task buffer pool each time under the CSDC-OS1. Then, the MCV returns to the service station and waits for the adaptive buffer pool to be full again after the service is completed. The size of the adaptive task buffer pool is very small compared to the number of e-bikes, which also results in fewer charging services per round under CSDC-OS1 compared to CSDC-OS2. After the task buffer pool is quickly full, the MCV starts to perform the charging task again, increasing the length of the MCV moving path.

As shown in Figure 8, the charging time of the MCV increases as the number of e-bikes increases because the increase in the number of e-bikes will increase the number of charging requests. Regardless of the number of e-bikes, CSDC-OS2 has a higher charging time than CSDC-OS1, which is because the CSDC-OS2 starts according to the e-bike energy directly. Once the ALERT message is received, the MCV will start. CSDC-OS2 does not consider the size of the task buffer pool. The received charging requests can be placed in the task buffer pool, so that the number of e-bikes served by MCV in each round of charging is more than CSDC-OS1. Thus, the MCV charging time is higher than CSDC-OS1 under the CSDC-OS2.

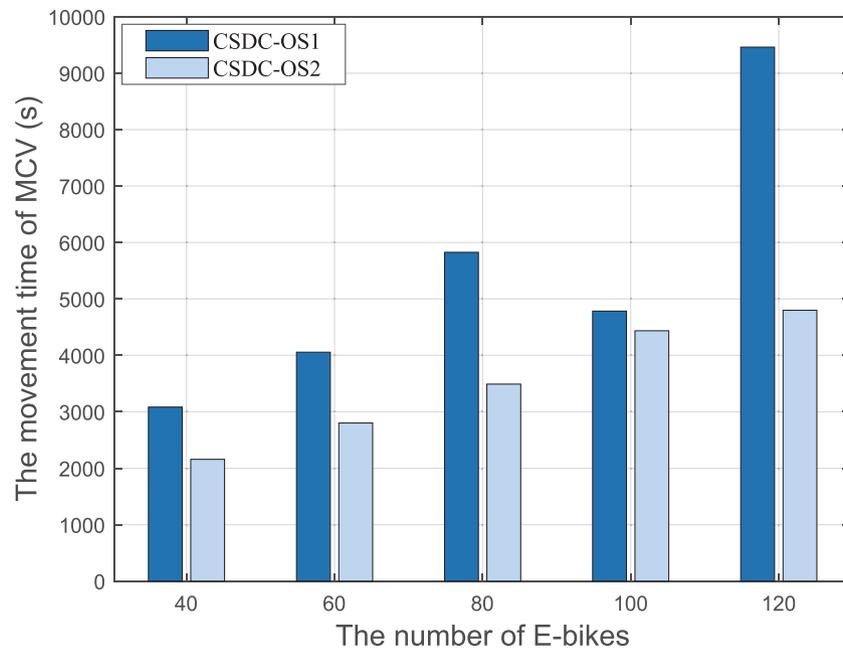


FIGURE 7 MCV movement time under different numbers of e-bikes

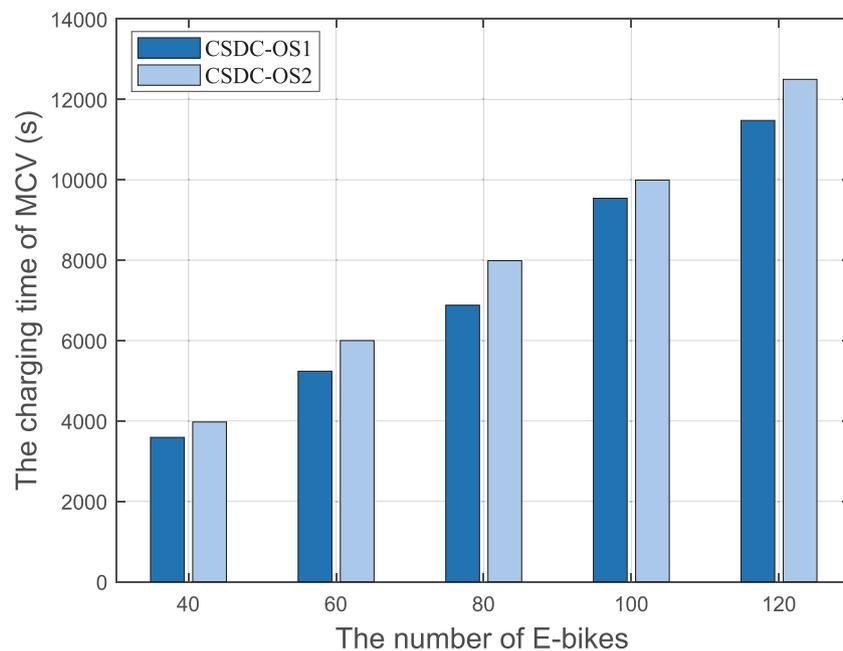


FIGURE 8 MCV charging time under different numbers of e-bikes

8.3 | Comparative analysis of MCV throughput

The MCV throughput is defined as the number of e-bikes that are successfully charged at different times as the simulation time changes. The higher the MCV charging throughput means that the less time the MCV is used for movement at the current moment and the higher the charging efficiency. As shown in Figure 9, it can be seen that under the CSDC-OS scheme, the MCV throughput increases with time and the throughput of the CSDC-OS2 is greater than that of the CSDC-OS1. This is because the number of e-bikes that are serviced by MCV in each charging round under CSDC-OS1 is determined by the size of the adaptive task buffer pool and the number of e-bikes that satisfy the conditions for insertion. However, the CSDC-OS2 does not consider the size of the task buffer pool. Under CSDC-OS2, the

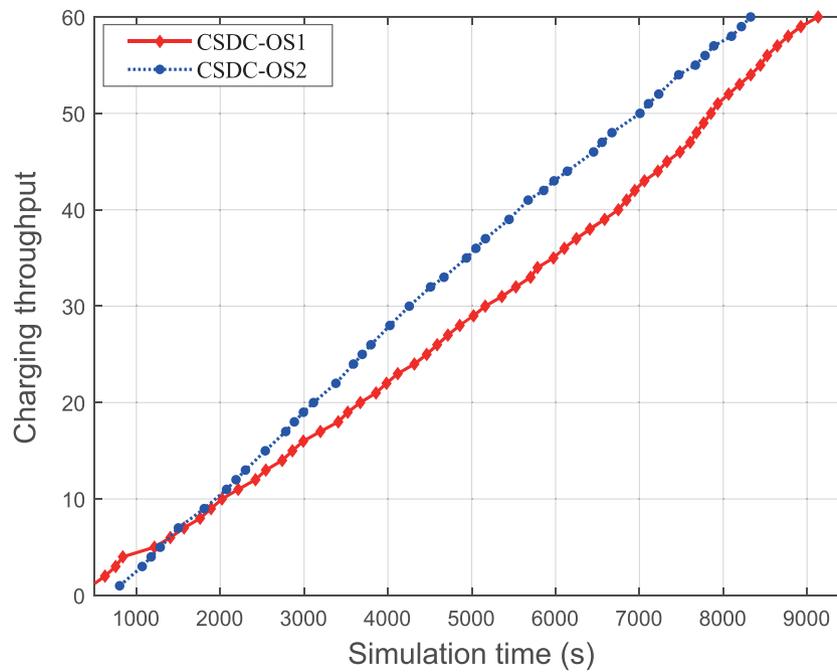


FIGURE 9 Comparison of throughput with simulation time

MCV starts to perform charging task when the ALERT message is received. The charging request can still be received during the MCV service as long as the charging schedulability condition is met and the MCV can serve it. In CSDC-OS2, the number of e-bikes that the MCV departs for each round is determined by the number of received charging requests, which is greater than the number of services served in CSDC-OS1. Therefore, the throughput of CSDC-OS2 is greater than the throughput in CSDC-OS1.

8.4 | Impact analysis of obstacles

Due to the complex operating environment of the shared e-bikes, the presence of obstacles not only affects the movement of the e-bikes in the network but also affects the charging path of the MCV. As shown in Figure 10, the Euclidean path length is the path length calculated according to the Euclidean distance, which means the straight-line distance between the actual moving path is the actual moving path length of the MCV in the network scene. As the number of obstacles increases, the actual moving path length and the Euclidean path length increase. Due to the presence of obstacles, the actual moving path length of the MCV is greater than the Euclidean path length. When the number of obstacles is zero, the Euclidean path length is not equal to the actual path length because we divide the two-dimensional network space into a grid with a side length of 1 m when using the A-star algorithm to determine the MCV moving path. The MCV moves along the diagonal or side length of the mesh, so the actual moving path length of the MCV is also greater than the Euclidean path length when there is no obstacle.

8.5 | Comparative analysis of MCV charging efficiency

Charging efficiency is one of the important indicators to measure the performance of the charging scheduling algorithm. In this paper, we define it as the ratio of the energy that the MCV uses to replenish the energy to the e-bike and the total energy spent completing the charging schedule during the entire charging schedule. The total energy spent by the MCV in completing the charging schedule is the sum of the energy consumed by the charging and the energy consumed by the movement. The higher charging efficiency means that the MCV spends more energy on energy replenishment and the performance of the entire charging system is better. We compare CSDC-OS with NJNP,²⁴ TADP,²⁵ and RCSS.³⁶ The NJNP charging scheme means that the MCV preferentially selects the space closest to its nearest node for

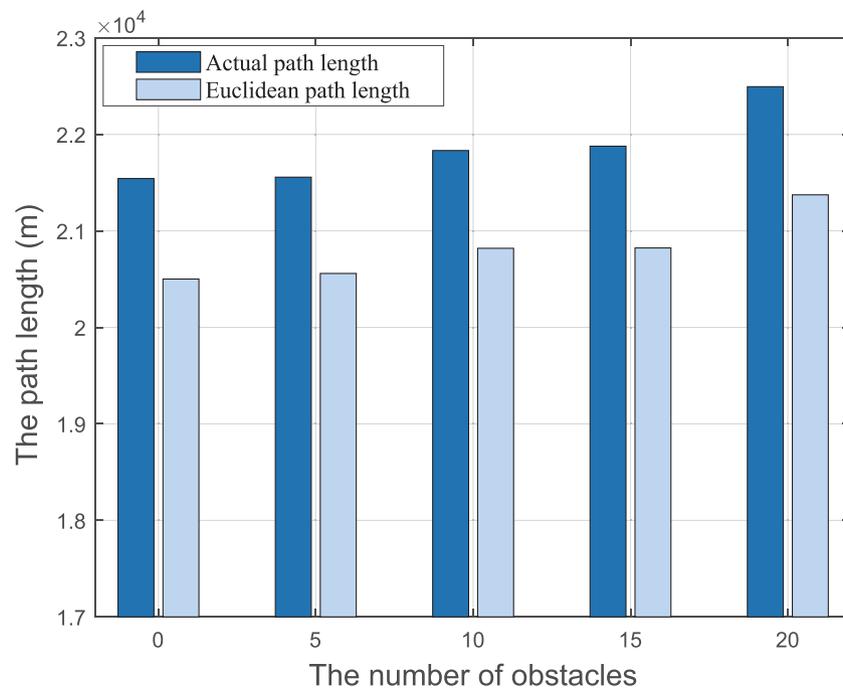


FIGURE 10 The path length is affected by the number of obstacles under CSDC-OS1

service. The TADP charging scheme not only considers the spatial characteristics of the node distribution but also considers the residual energy of the node to ensure that the MCV preferentially selects the node with less residual energy and closer to the MCV. The RCSS charging scheme comprehensively considers the distance from the node to the MCV, the remaining energy, and the energy consumption rate of the node to ensure that the MCV preferentially serves each node with a short distance to the MCV, less residual energy, and a larger energy consumption rate.

As shown in Figure 11, it can be seen that the CSDC-OS scheme is higher in charging efficiency than the other three charging schemes. This is because the CSDC-OS fully considers the number of charging requests in the current network and the energy consumption of the e-bike and dynamically adjusts the size of the task buffer pool and the value of E_L . In addition, CSDC-OS introduces e-bike selective insertion algorithm to ensure that the MCV can service more e-bikes during each round of charging. When determining the charging order, the CSDC-OS fully considers the frequency of use, the remaining energy, and the distance to the MCV and then performs a combination optimization to determine the charging order by the linear skyline query method. Compared with the single influencing factors, this method is more suitable for the complicated operating environment of the e-bike, thereby increasing the number of services provided for e-bikes and reducing the movement of MCV.

8.6 | Comparative analysis of average response duration

The average response duration is defined as the time interval from when the e-bike issues a charging request to the time that the e-bike is responded by the MCV and divides by the number of e-bikes that are all requested to be charged, which reflects the average waiting time of the e-bikes before being responded to. The smaller the average response duration, the shorter the charging delay and the lower the probability that the e-bike will not work due to energy exhaustion. The average response duration is mainly affected by the MCV starting mechanism. To demonstrate the performance of the two departure mechanisms proposed by CSDC-OS, we consider comparing CSDC-OS1 with adaptive task buffer pool to CSDC-OS1com (cf. CSDC-OS1 and CSDC-OS1com) based on fixed task buffer pool (set to 10). We also consider the CSDC-OS2 with dual-energy threshold and the charging scheme based on fixed charging threshold (set to 225 J) CSDC-OS2com (cf. CSDC-OS2 and CSDC-OS2com) for comparison. As shown in Figure 12, in the case of small-scale e-bikes, $t(\text{CSDC-OS2}) < t(\text{CSDC-OS1}) < t(\text{CSDC-OS2com}) < t(\text{CSDC-OS1com})$ in terms of the average response duration. The average response duration is longer than CSDC-OS2 for CSDC-OS1 and CSDC-OS1com because the charging buffer is full for a long time and charging delay is long when the charging request is small. The adaptive

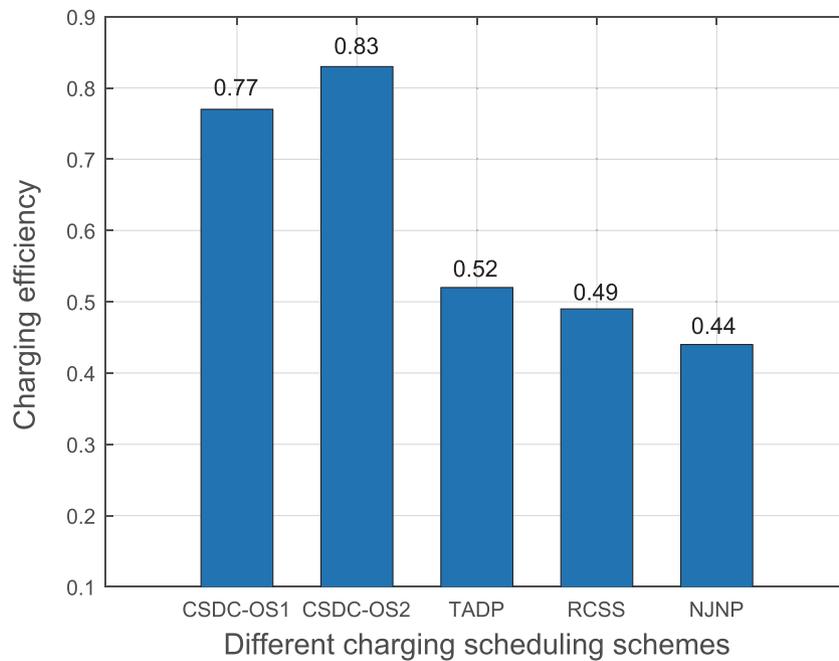


FIGURE 11 Comparison of charging efficiency under different charging scheduling mechanisms

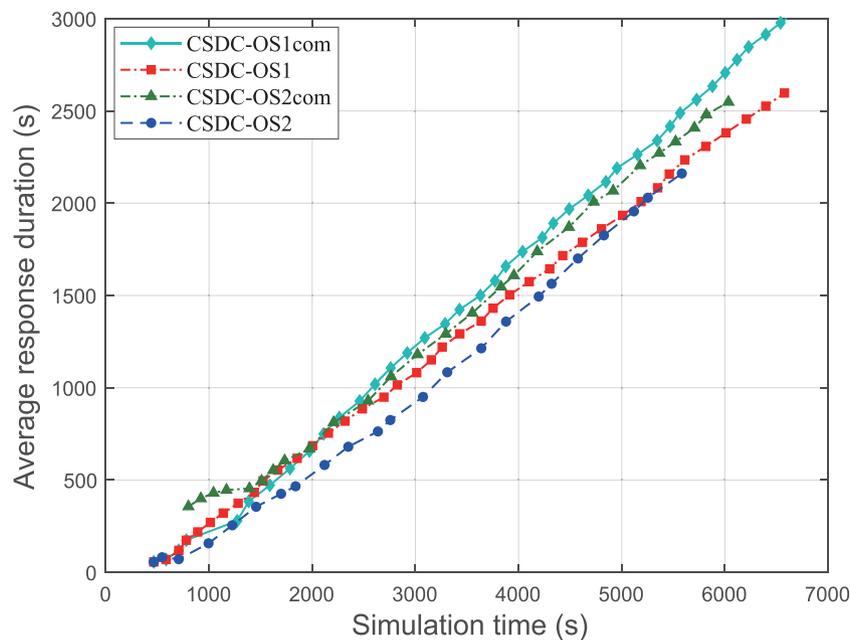


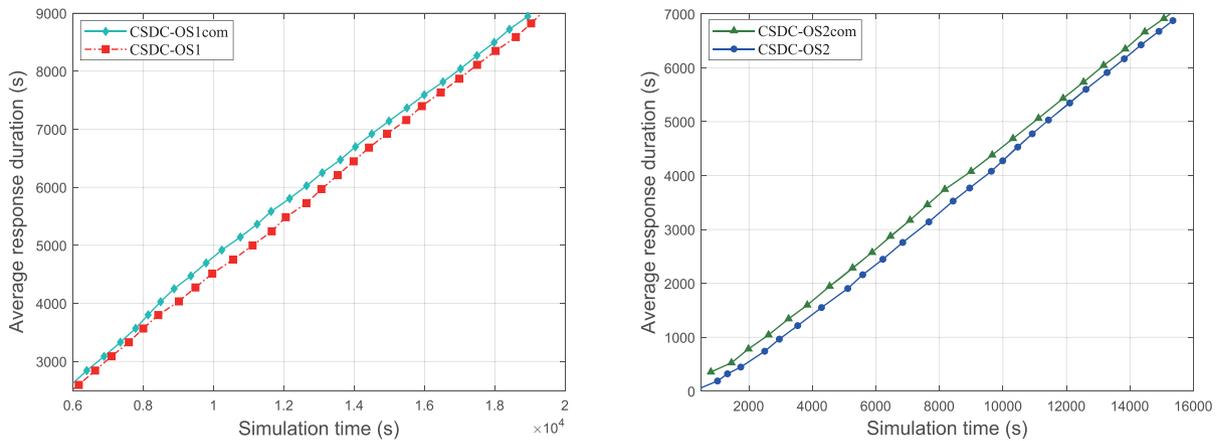
FIGURE 12 Comparison of average response duration of 40 e-bikes

task buffer pool adaptively adjusts the size of the task buffer pool according to the current number of unserved charging requests. Compared with CSDC-OS1com, the buffer pool size is closer to the current number of e-bikes that need to be charged. Therefore, $t(\text{CSDC-OS1}) < t(\text{CSDC-OS1com})$. Compared with CSDC-OS2com, CSDC-OS2 reduces the length of time that MCV spends on mobile in the case of fewer charging requests, thus reducing the average response duration.

As shown in Figure 13, under the large-scale number of e-bikes, the average response duration change trend under different starting mechanisms is unchanged. However, the difference in the average response time of different departure mechanisms is gradually reduced. We divide it into two figures so that the difference is more clearly seen. In Figure 13A, it can be seen that the average response duration difference between CSDC-OS1com and CSDC-OS1 is

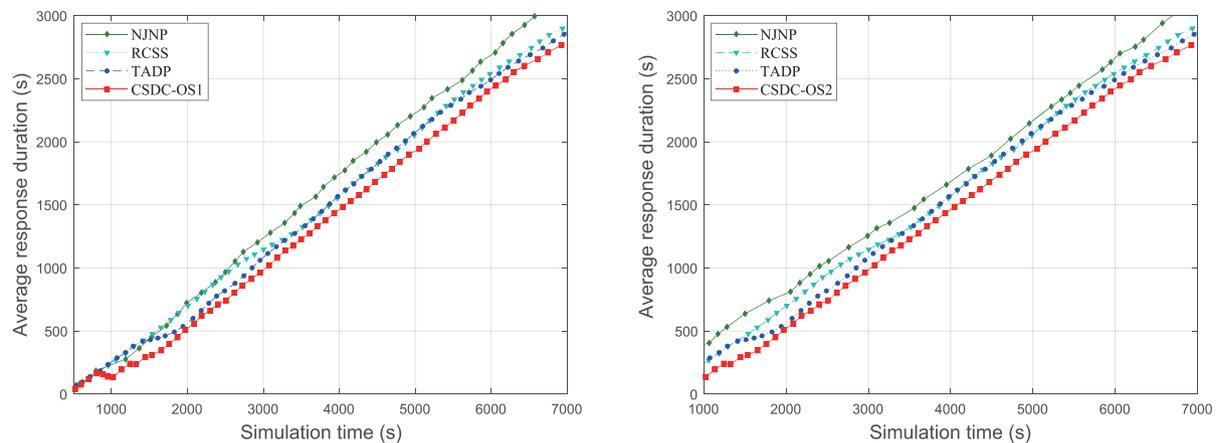
small and CSDC-OS1 is slightly better than CSDC-OS1com. This is because the number of e-bike increases leads to the number of charging requests collected by the MCV increases. Whether the fixed task buffer pool or the adaptive task buffer pool is easier to meet the MCV departure conditions, thus the gap is reduced. However, the starting mechanism under the adaptive task buffer pool is more “sensitive” to the charging request. Its size automatically changes with the number of currently unserved e-bikes, making their average response duration slightly better than CSDC-OS1com. In Figure 13B, it can be seen that CSDC-OS2 is superior to CSDC-OS2com in average response duration, but the gap is reduced compared to the small-scale number of e-bikes. This is because the number of charging requests collected by the MCV has increased. In CSDC-OS2, the advantage of MCV to reduce the duration of movement is reduced, thereby reducing the advantage of reducing the average response time.

As shown in Figure 14, we compare the performance differences between CSDC-OS, NJNP, TADP, and RCSS in terms of charging response duration. It can be seen that both CSDC-OS1 and CSDC-OS2 are superior to the other three charging scheduling methods. This is because NJNP, TADP, and RCSS all adopt the on-demand charging scheduling method under the fixed task buffer pool. As shown in Figure 14A, CSDC-OS1 is more likely to satisfy the MCV starting condition when the MCV receives less charging request, which is similar to the reason described above with reference to Figures 12 and 13A. As shown in Figure 14B, as the simulation time increases, the average response duration of CSDC-OS2 is better than the other three charging scheduling methods. The CSDC-OS2 is a charging scheme based on the MCV dual-energy threshold starting mechanism. Compared with the charging scheme based on the fixed task



(A) Comparison of the starting mechanism of the adaptive task buffer pool. (B) Comparison based on the dual threshold starting mechanism.

FIGURE 13 Comparison of the average response duration of 120 e-bikes



(A) Comparison of the starting mechanism of the adaptive task buffer pool. (B) Comparison based on the dual threshold starting mechanism.

FIGURE 14 Comparison of the average response duration of 60 e-bikes

buffer size, this starting mechanism based on the dual-energy threshold of the e-bike can directly reflect the energy dynamic process of e-bikes and is more suitable for the complex operating environment, so the performance of CSDC-OS2 is better.

In summary, CSDC-OS2 is superior to CSDC-OS1 in reducing the average response time. As the number of e-bikes increases, the gap between CSDC-OS1 and CSDC-OS2 gradually narrows. This is because as the number of e-bikes increases, the number of charging requests received by the MCV increases and the CSDC-OS1 is easy to meet the MCV starting mechanism. However, the dual-energy threshold-based starting mechanism can directly reflect the energy change process of e-bikes, so CSDC-OS2 is slightly better than CSDC-OS1.

8.7 | Comparative analysis of average service duration

The average service duration is defined as the sum of the time when the MCV responds to the charging request and the time when the MCV moves to serve the e-bikes at the current time, and then the sum value is divided by the number of serviced e-bikes. The smaller the average service duration, the shorter the waiting time for the remaining unpaid e-bikes and the lower the probability that other e-bikes will be unusable due to energy exhaustion. As shown in Figure 15, the average service duration of CSDC-OS2 and CSDC-OS2com is gradually lower than CSDC-OS1 and CSDC-OS1com with the simulation time under the small-scale number of e-bikes. In this case, the MCV collects fewer charging requests, which will cause the fixed task buffer pool (CSDC-OS1com) and the adaptive task buffer pool (CSDC-OS1) to be full for a long time. At the same simulation time, the more number of e-bike services, the shorter the average response duration and the longer waiting time under CSDC-OS1 and CSDC-OS1com. Thus, the number is less than CSDC-OS2, and the average service duration is higher than CSDC-OS2.

As shown in Figure 16, the average service duration change trend under different starting mechanisms is basically unchanged compared with Figure 15. However, the CSDC-OS2 and CSDC-OS1 gradually lose their advantages in reducing the average service duration. For CSDC-OS1, the MCV will receive a large number of charging requests when the number of e-bikes increases. Compared with small-scale quantities, MCV can easily meet the starting mechanism conditions and reduce the waiting time for e-bikes, which makes the number of service e-bikes gradually increase under the same simulation time. For CSDC-OS2, the charging request is frequent as the number of e-bikes increases. After the MCV starts, it can serve more e-bikes under the condition of ensuring charging schedulability, which gradually reduces average service duration with the simulation time.

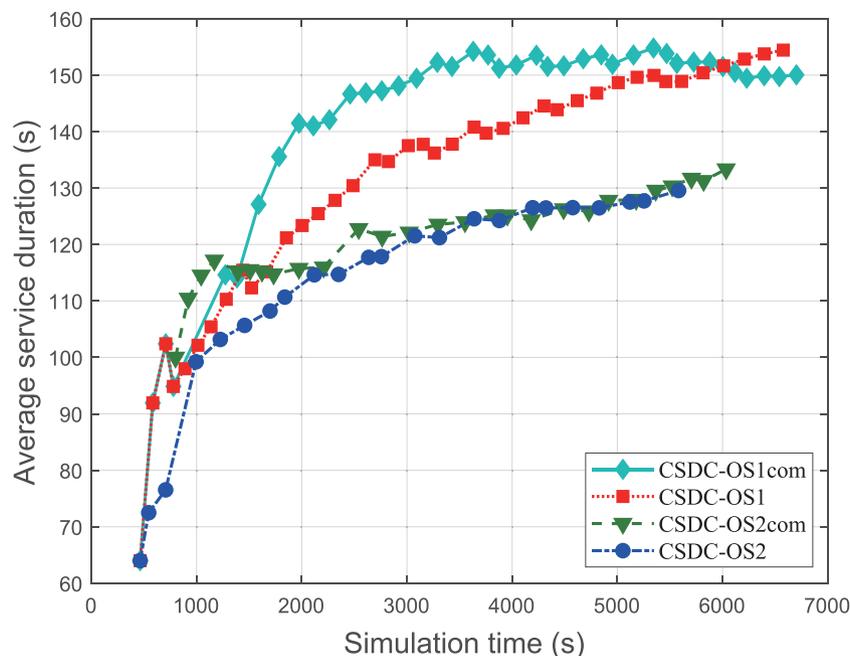


FIGURE 15 Comparison of average service durations for 40 e-bikes

8.8 | Comparative analysis of MCV moving path length

The length of the MCV moving path greatly affects the charging throughput. As shown in Figure 17, the length of the MCV moving path under different charging mechanisms shows that the CSDC-OS scheme proposed in this paper significantly reduces the MCV moving path length compared with NJNP, TADP, and RCSS. This is because NJNP, TADP, and RCSS are more suitable for static node topology scenarios. On the one hand, CSDC-OS uses the selective insertion algorithm to increase the service throughput without approximating the length of the MCV moving path. For NJNP, TADP, and RCSS, these schemes need to determine the charging execution process based on the received charging

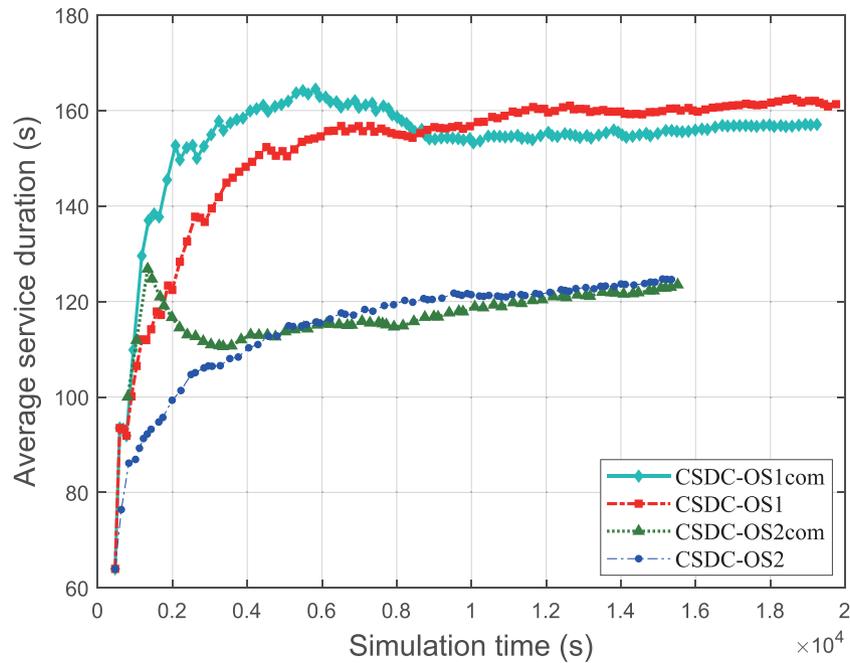


FIGURE 16 Comparison of average service durations for 120 e-bikes

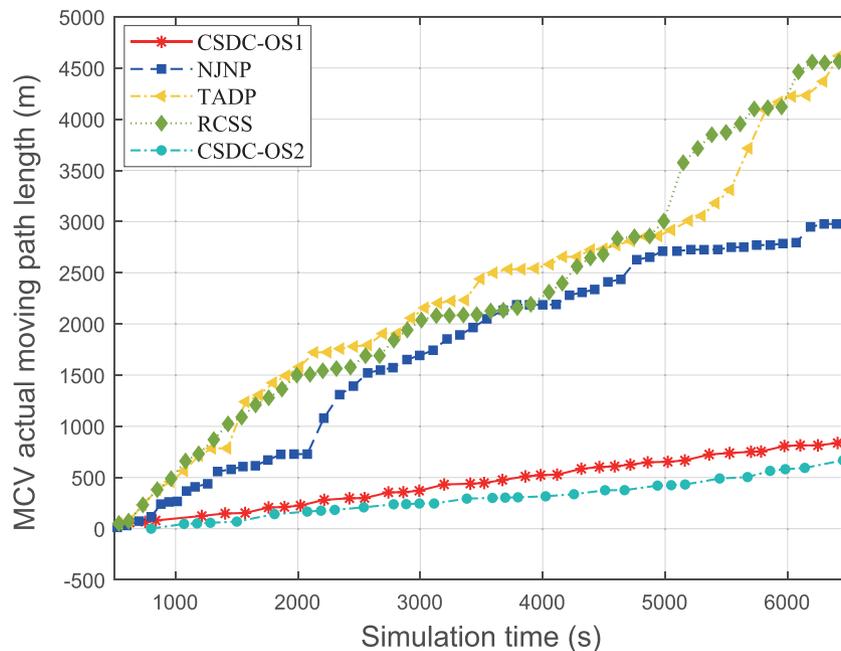


FIGURE 17 The moving length of MCV under different charging mechanisms

request and metrics, regardless of the situation in which the e-bike is inserted into the charging sequence priority service. This problem leads to the MCV repeatedly fold back in the network, increasing the length of the moving path of the MCV and wasting resources.

On the other hand, the CSDC-OS performs charging tasks according to the MCV starting mechanism of the dynamic change of the e-bike. The MCV starts to perform charging tasks when the size of the adaptive task buffer pool is full or the ALERT message is received. Under the condition that the charge schedulability is satisfied, the charging request can be continuously processed. In the NJNP, TADP, and RCSS, the MCV starts to serve the e-bike after the fixed task buffer pool is full. In NJNP, TADP, and RCSS, MCV returns to the service station to wait for the next round of services only after completing a round of services. In a shared e-bike operating environment where charging requests are frequent, there may be a new charging demand immediately after the MCV returns to the service station, which will increase the MCV moving distance and reduce the MCV charging efficiency.

9 | CONCLUSIONS

In this paper, we propose a charging scheduling scheme named CSDC-OS based on the dynamic change of shared e-bikes in obstacle space. Firstly, the MCV starting mechanism based on the dynamic change of e-bikes can effectively balance the charging load in different environments and reduce charging delay and resource waste. Then, we make a theoretic analysis on the charging schedulability conditions of MCVs to ensure that MCVs can successfully return to the service station when they need to charge. The e-bike charging order determination algorithm based on linear skyline query can take various characteristics of shared e-bikes into account to select the best service order, which can meet the charging demand and ensure the relative superiority of the overall solution. In addition, we study the conditions of the e-bike pluggable charging queue in theory. The conclusions have important guiding significance for designing MCV charging paths in complex environments. This makes the whole scheme closer to real-life scenarios and improves the utilization of MCV resources. Experimental results show that compared with other algorithms, the CSDC-OS improves response duration, MCV moving path length, and MCV charging efficiency by at least 8%, 40%, and 32% on average, respectively.

ACKNOWLEDGMENTS

The work described in this paper was supported by the grant from the Natural Science Foundation of Hunan Province (No. 2018JJ3692) and National Natural Science Foundation of China (No. 61402542), and the Fundamental Research Funds for Central Universities of the Central South University (China) under Grant Nos. 2021zzts0735 and 2021zzts0753.

ORCID

Ping Zhong  <https://orcid.org/0000-0003-3393-8874>

REFERENCES

1. Meddin R. The bike-sharing world map. <http://www.bikesharingmap.com>. Accessed September 17, 2019; 2019.
2. CAAM. Chinese Association of Automobile Manufacturers—automotive statistics. <http://www.caam.org.cn/english/newslis/a101-1.html>. Accessed September 17, 2019; 2015.
3. CAAM. Chinese Association of Automobile Manufacturers—motorcycle statistics. <http://www.caam.org.cn/english/newslis/a104-1.html>. Accessed September 17, 2019; 2015.
4. Bai L, Liu P, Chen Y, Zhang X, Wang W. Comparative analysis of the safety effects of electric bikes at signalized intersections. *Transport Res D-TR E*. 2013;20:48-54.
5. Fishman E, Washington S, Haworth N, Watson A. Factors influencing bike share membership: an analysis of Melbourne and Brisbane. *Transport Res A-POL*. 2015;71:17-30.
6. Ji S, Cherry CR, Han LD, Jordan DA. Electric bike sharing: simulation of user demand and system availability. *J Clean Prod*. 2014;85:250-257.
7. Fyhri A, Fearnley N. Effects of e-bikes on bicycle use and mode share. *Transport Res D-TR E*. 2015;36:45-52.
8. Florez D, Carrillo H, Gonzalez R, Herrera M, Hurtado-Velasco R, Cano M, Roa S, Manrique T. Development of a bike-sharing system based on pedal-assisted electric bicycles for Bogota City. *Electr*. 2018;7(11):337.
9. Samet B, Couffin F, Zolghadri M, Barkallah M, Haddar M. Performance analysis and improvement of the bike sharing system using closed queuing networks with blocking mechanism. *Sustainab*. 2018;10(12):4663.

10. Chiariotti F, Pielli C, Zanella A, Zorzi M. A dynamic approach to rebalancing bike-sharing systems. *Sensors*. 2018;18(2):512.
11. Datner S, Raviv T, Tzur M, Chemla D. Setting inventory levels in a bike sharing network. *Transport Sci*. 2019;53(1):62-76.
12. Liu Y, Tian L. A graded cluster system to mine virtual stations in free-floating bike-sharing system on multi-scale geographic view. *J Clean Prod*. 2020;281:124692.
13. Vishkaei BM, Mahdavi I, Mahdavi-Amiri N, Khorram E. Balancing public bicycle sharing system using inventory critical levels in queuing network. *Compu Ind Eng*. 2020;141:106277.
14. Zhao Y-X, Su Y-S, Chang Y-C. A real-time bicycle record system of ground conditions based on Internet of Things. *IEEE Access*. 2017;5:17525-17533.
15. Corno M, Roselli F, Savaresi SM. Bilateral control of SeNZA—a series hybrid electric bicycle. *IEEE T Contr Syst T*. 2016;25(3):864-874.
16. Cahyono SI, Anwar M, Diharjo K, Triyono T, Hapid A, Kaleg S. Finite element analysis of electric bicycle frame geometries. In: AIP Conference Proceedings, Vol. 1788 AIP Publishing LLC; 2017:30084.
17. Guanetti J, Formentin S, Corno M, Savaresi SM. Optimal energy management in series hybrid electric bicycles. *Automatica*. 2017;81:96-106.
18. Hung NB, Lim O. A review of history, development, design and research of electric bicycles. *Appl Energ*. 2020;260:114323.
19. Lin J, Schofield N, Emadi A. External-rotor 6 – 10 switched reluctance motor for an electric bicycle. *IEEE T Transp Electr*. 2015;1(4):348-356.
20. Kheirandish A, Motlagh F, Shafiabady N, Dahari M, Wahab AKA. Dynamic fuzzy cognitive network approach for modelling and control of PEM fuel cell for power electric bicycle system. *Appl Energ*. 2017;202:20-31.
21. Adhisuwignjo S, Siradjuddin I, Rifa'i M, Putri RI. Development of a solar-powered electric bicycle in bike sharing transportation system, IOP Conference Series: Earth and Environmental Science. *IOP Publishing*. 2017;70(1):12025.
22. Wang W, Malysz P, Khan K, Gauchia L, Emadi A. Modeling, parameterization, and benchmarking of a lithium ion electric bicycle battery. In: 2016 IEEE Energy Conversion Congress and Exposition (ECCE) IEEE; 2016:1-7.
23. Howey B, Bilgin B, Emadi A. Design of an external-rotor direct drive e-bike switched reluctance motor. *IEEE T Veh Technol*. 2020;69(3):2552-2562.
24. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things*. 2017;4(5):1125-1142.
25. Verma G, Sharma V. A novel thermoelectric energy harvester for wireless sensor network application. *IEEE T Ind Electron*. 2018;66(5):3530-3538.
26. Ahmad A, Alam MS, Chabaan R. A comprehensive review of wireless charging technologies for electric vehicles. *IEEE T Transp Electr*. 2017;4(1):38-63.
27. Feng Y, Wang Y, Zheng H, Mi S, Tan J. A framework of joint energy provisioning and manufacturing scheduling in smart industrial wireless rechargeable sensor networks. *Sensors*. 2018;18(8):2591.
28. He L, Kong L, Gu Y, Pan J, Zhu T. Evaluating the on-demand mobile charging in wireless sensor networks. *IEEE T Mobile Comput*. 2014;14(9):1861-1875.
29. Lin C, Wang Z, Han D, Wu Y, Yu CW, Wu G. TADP: enabling temporal and distant priority scheduling for on-demand charging architecture in wireless rechargeable sensor networks. *J Syst Architect*. 2016;70:26-38.
30. Liang W, Xu Z, Xu W, Shi J, Mao G, Das SK. Approximation algorithms for charging reward maximization in rechargeable sensor networks via a mobile charger. *IEEE ACM T Network*. 2017;25(5):3161-3174.
31. Lin C, Han D, Deng J, Wu G. P²S: a primary and passer-by scheduling algorithm for on-demand charging architecture in wireless rechargeable sensor networks. *IEEE T Veh Technol*. 2017;66(9):8047-8058.
32. Lin C, Wei S, Deng J, Obaidat MS, Song H, Wang L, Wu G. GTCCS: a game theoretical collaborative charging scheduling for on-demand charging architecture. *IEEE T Veh Technol*. 2018;67(12):12124-12136.
33. Lin C, Wu Y, Liu Z, Obaidat MS, Yu CW, Wu G. GTCharge: a game theoretical collaborative charging scheme for wireless rechargeable sensor networks. *J Syst Software*. 2016;121:88-104.
34. Lin C, Zhou Y, Song H, Yu CW, Wu G. OPPC: an optimal path planning charging scheme based on schedulability evaluation for WRSNs. *ACM T Embed Comput S*. 2017;17(1):1-25.
35. Lin C, Wang K, Chu Z, Wang K, Deng J, Obaidat MS, Wu G. Hybrid charging scheduling schemes for three-dimensional underwater wireless rechargeable sensor networks. *J Syst Software*. 2018;146:42-58.
36. Wang C, Li J, Yang Y, Ye F. A hybrid framework combining solar energy harvesting and wireless charging for wireless sensor networks. In: IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications IEEE; 2016:1-9.
37. Liu W, Tang Y, Yang F, Dou Y, Wang J. A multi-objective decision-making approach for the optimal location of electric vehicle charging facilities. *Comput Mater Continua*. 2019;60(2):813-834.
38. Chen Z, Chen X, Zhang D, Zeng F. Collaborative mobile charging policy for perpetual operation in large-scale wireless rechargeable sensor networks. *Neurocomput*. 2017;270:137-144.
39. Liu T, Wu B, Wu H, Peng J. Erratum to “Low-cost collaborative mobile charging for large-scale wireless sensor networks”. *IEEE T Mobile Comput*. 2017;16(11):3278-3278.
40. Liang W, Xu W, Ren X, Jia X, Lin X. Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles. *ACM T Sensor Netw*. 2016;12(2):1-26.

41. Zhong P, Zhang Y, Ma S, Kui X, Gao J. RCSS: a real-time on-demand charging scheduling scheme for wireless rechargeable sensor networks. *Sensors*. 2018;18(5):1601.
42. Fu L, Cheng P, Gu Y, Chen J, He T. Optimal charging in wireless rechargeable sensor networks. *IEEE T Veh Technol*. 2015;65(1):278-291.
43. Lin C, Sun Y, Wang K, Chen Z, Xu B, Wu G. Double warning thresholds for preemptive charging scheduling in wireless rechargeable sensor networks. *Comput Netw*. 2019;148:72-87.
44. Ehrgott M. *Multicriteria Optimization*, Vol. 491: Springer Science & Business Media; 2005.
45. Papadias D, Tao Y, Fu G, Seeger B. An optimal and progressive algorithm for skyline queries. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data; 2003:467-478.

How to cite this article: Zhong P, Xu A, Gao J, Zhang Y, Chen Y. An efficient on-demand charging scheduling scheme for mobile charging vehicle. *Int J Commun Syst*. 2021;34:e4919. <https://doi.org/10.1002/dac.4919>