# Efficient Schedule of Path and Charge for a Mobile Charger to Improve Survivability and Throughput of Sensors with Adaptive Sensing Rates

You-Chiun Wang and Yu-Cheng Bai

**Abstract**—Wireless sensor networks provide long-term monitoring of the environment, but sensors are powered by small batteries. Using a *mobile charger (MC)* to replenish energy of sensors is one promising solution to prolong their usage time. Many approaches have been developed to find the MC's moving path, and they assume that sensors have a fixed *sensing rate (SR)* and prefer to fully charge sensors. In practice, sensors can adaptively adjust their SRs to meet application demands or save energy. Besides, due to the fully charging policy, some sensors with low energy may take long to wait for the MC's service. Thus, the paper formulates a *path and charge (P&C) problem*, which asks how to dispatch the MC to visit sensors with adaptive SRs and decide their charging time, such that both survivability and throughput of sensors can be maximized. Then, we propose an *efficient P&C scheduling (EPCS) algorithm*, which builds the shortest path to visit each sensor. To make the MC fast move to charge the sensors near death, some sensors with enough energy are excluded from the path. Moreover, EPCS adopts a floating charging mechanism based on the ratio of workable sensors and their energy depletion. Simulation results verify that EPCS can significantly improve the survivability and throughput of sensors.

**Index Terms**—mobile charger, path & charge scheduling, sensing rate, survivability, wireless sensor network.

## 1 Introduction

The Internet of things has ushered in a brand-new era, where *wireless sensor networks (WSNs)* are extensively used in industry and people's livelihood [1], [2]. A WSN is made up of many sensors, which are small wireless devices that can gather data from the surroundings and report what they collect to a remote sink. Various WSN applications have been proposed to improve the quality of life, from air-pollution detection [3] to health care [4], light control [5], object surveillance [6], precision agriculture [7], and smart shopping [8].

In most WSN applications, the sensors have to provide long-term monitoring. As limited by their size, sensors can only adopt small batteries to be the power supply. The feasible solutions to this dilemma include energy-efficient routing [9], data compression [10], sleep scheduling [11], energy harvesting [12], and wireless charging [13]. This paper aims to use a *mobile charger (MC)* to extend the available working time of sensors, which is a wireless charger equipped on one mobile platform (e.g., vehicle or drone) [14] and can move to visit sensors for recharging their batteries.

How to arrange the MC's path to visit sensors and their charging time has a great impact on the available working time of sensors. The existing methods usually consider that sensors have the identical *sensing rate (SR)*. Some methods even assume that the MC can fast charge sensors' batteries, so they take the fully charging policy. In real applications, sensors may have *adaptive* SRs, in the sense that their SRs will change in different situations. Specifically, when sensors detect abnormal events or reside in an interested region, they may be asked to increase their SRs to collect more data [15]. On the other hand, the sensors whose residual energy become low could reduce their SRs to conserve energy [16]. Adaptive SRs lead to different speeds of energy depletion for sensors, which degrades the performance of those methods based on the assumption of fixed SRs. Moreover, when the MC simply takes the fully charging policy, the waiting time for each sensor to be served by the MC will rise accordingly. Some low-energy sensors may stop working quickly (due to no energy) and remain in the "dead" state for a long time (until the MC recharges them). These sensors not only leave coverage holes but also decrease the WSN's throughput.

This paper formulates a *path and charge (P&C) problem* to schedule both moving path and charging time for the MC with the consideration of adaptive SRs of sensors. The objective is to increase the percentage of workable sensors (i.e., survivability) and the amount of sensing data sent to the sink (i.e., throughput). Therefore, we propose an *efficient P&C scheduling (EPCS) algorithm*, whose idea is to let the MC charge the sensors in urgent need of energy as soon as possible. To do so, EPCS first finds the shortest path to visit all sensors, and then removes some sensors whose energy is enough for operation from the path. Depending on the proportion and energy consumption of workable sensors, the MC will charge sensors in a floating fashion. Through simulations, we demonstrate that our EPCS algorithm can keep high survivability and throughput of sensors, as compared with the existing methods.

## 2 Related Work

How to place wireless chargers to extend a WSN's lifetime is widely discussed. The work [17] uses the fewest chargers to recharge all sensors and finds their optimal locations by the Daubechies wavelet method. The study [18] analyzes the effect of obstacles on charger placement, and finds their locations by a dominating coverage set extraction scheme. Dai et al. [19] consider placing chargers to ensure electromagnetic radiation safety in the sensing field, and convert the placement problem to a multi-dimensional 0/1 knapsack problem. Given a set of

*The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: ycwang@cse.nsysu.edu.tw; m073040018@student.nsysu.edu.tw).*

directional chargers, the study [20] finds the best location and orientation angle for each charger subject to the connectivity constraint, such that the overall charging utility is maximized. Other issues of wireless chargers are also addressed. Guo et al. [21] propose a concurrent charging schedule to mitigate interference between two chargers and save the time to recharge sensors. The work [22] moves mobile sensors to the locations of events for analysis. Mobile sensors will visit chargers on the way to event locations, so as to replenish energy during their working time. However, these studies consider only fixed chargers.

The problem of finding the MC's optimal moving path to charge sensors is NP-hard [23]. Many studies [24], [25], [26], [27] are based on approximate solutions to the *traveling salesman problem (TSP)*, which builds a Hamiltonian cycle to visit sensors. Various strategies are also developed. The *first-come-first-serve (FCFS)* strategy [28] makes the MC charge sensors based on the sequence of their charging requests. For the *nearest-job-next with preemption (NJNP)* strategy [29], the MC picks the closest requestor as the next candidate for charging. In the *earliest-deadline-first (EDF)* strategy [30], the MC first serves those sensors about to exhaust energy. The *temporal and distantial priority charging scheduling (TADP)* method [31] considers the MC's moving distance and also the arrival time of requests from sensors. In [32], a Hilbert curve is used to find the MC's path to visit each sensor. However, the above studies assume that sensors have the same SR. This motivates us to develop the EPCS algorithm to charge sensors with adaptive SRs efficiently, so as to improve their survivability and throughput.

## 3 System Model

We are given a set $\hat{\mathcal{S}}$ of sensors in the sensing field. Each sensor $s_i \in \hat{\mathcal{S}}$ is powered by a rechargeable battery whose capacity is $e_i^{\max}$, and $e_i$ denotes $s_i$'s residual energy. As discussed in Section 3.1, $s_i$ spends energy on sensing, sending, and receiving data. The SR $r_i$ of $s_i$ may dynamically change (i.e., adaptive SR). Specifically, if $e_i$ is below a threshold, $s_i$ switches to the low-power mode, which decreases $r_i$ for energy saving (the detail is given in Section 3.2). However, $s_i$ stops working if $e_i < e_i^{\min}$, where $e_i^{\min}$ is the minimum required energy for $s_i$ to operate, until $s_i$'s battery is recharged.

There is one MC moving in the sensing field (with a constant speed) to recharge sensors round by round. In each round, the MC visits a subset of sensors in $\hat{\mathcal{S}}$ for charging according to the proposed algorithm. We have to gather the parameters of each sensor to execute the algorithm, including its position, residual energy $e_i$, and SR $r_i$. Specifically, the positions of sensors can be known during the deployment stage or by using some positioning techniques [33]. On the other hand, sensors can periodically piggyback on sensing data to notify the sink of their $e_i$ and $r_i$ values. Notice that if the sensing field has obstacles, we can adopt the Dijkstra-based methods [34], [35] to find the shortest path for the MC to detour obstacles and reach its target locations.

### 3.1 Energy Expenditure and Replenishment

According to [36], the amount of energy for a sensor $s_i \in \hat{\mathcal{S}}$ to create one packet of sensing data, whose length is $\lambda$ bits, can be estimated as follows:

$$\tilde{E}_{\mathtt{se}}(s_i, \lambda) = (u_i^{\mathtt{se}} \times c_i^{\mathtt{se}} \times t_i^{\mathtt{se}}) \times \lambda. \quad (1)$$

TABLE 1: Summary of notations.

| Notation | Definition |
|---|---|
| $\hat{\mathcal{S}}$, $\hat{\mathcal{S}}_{\mathtt{A}}$ | Sets of all sensors and workable sensors |
| $e_i$ | Residual energy of a sensor $s_i \in \hat{\mathcal{S}}$ |
| $e_i^{\max}$ | Capacity of $s_i$'s battery |
| $e_i^{\min}$ | Minimum required energy for $s_i$ to operate |
| $r_i$ | SR of $s_i$ (high SR: $r_i^{\mathtt{H}}$, low SR: $r_i^{\mathtt{L}}$, threshold: $\delta$) |
| $u_i^{\mathtt{se}}$, $c_i^{\mathtt{se}}$, $t_i^{\mathtt{se}}$ | Voltage, current, and time for $s_i$ to produce a bit |
| $\zeta_i^{\mathtt{tx}}$, $\zeta_i^{\mathtt{am}}$ | Energy for $s_i$'s transmitter and amplifier to send a bit |
| $\zeta_i^{\mathtt{rx}}$ | Energy for $s_i$'s receiver to get a bit |
| $\mathcal{P}$ | Moving path ($\mathcal{P}_{\mathtt{SH}}$: shortest path to visit all sensors) |
| $v_j$ | A visited node in $\mathcal{P}$ ($v_0$: MC's POD) |
| $\tilde{E}_{\mathtt{mv}}$ | Energy taken by the MC to move a unit distance |
| $\mu_{\mathtt{avg}}$ | Average EER of workable sensors (threshold: $\mu_{\mathtt{th}}$) |
| $\Delta_{\mathtt{H}}$, $\Delta_{\mathtt{L}}$ | Two thresholds defined in the SNC module ($\Delta_{\mathtt{H}} > \Delta_{\mathtt{L}}$) |
| $\alpha$, $\beta$ | Two coefficients in Eq. (10) for partially charging |

In Eq. (1), $u_i^{\mathtt{se}}$, $c_i^{\mathtt{se}}$, and $t_i^{\mathtt{se}}$ are the amount of voltage, current, and time for $s_i$ to produce a bit, respectively. When $s_i$ sends the packet to a node $s_j$, $s_i$ spends an amount of energy:

$$\tilde{E}_{\mathtt{tx}}(s_i, s_j, \lambda) = [\zeta_i^{\mathtt{tx}} + \zeta_i^{\mathtt{am}} \times \tilde{D}(s_i, s_j)^2] \times \lambda, \quad (2)$$

where the transmitter and amplifier of $s_i$ spend $\zeta_i^{\mathtt{tx}}$ and $\zeta_i^{\mathtt{am}}$ energy to send a bit, respectively. In Eq. (2), $\tilde{D}(s_i, s_j)$ is the distance between $s_i$ and $s_j$, which is calculated by

$$\tilde{D}(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (3)$$

where the coordinates of $s_i$ and $s_j$ are $(x_i, y_i)$ and $(x_j, y_j)$, respectively. On the other hand, let $\zeta_j^{\mathtt{rx}}$ denote the amount of energy required by $s_j$'s receiver to get one bit. To receive the packet, $s_j$ spends an amount of energy:

$$\tilde{E}_{\mathtt{rx}}(s_j, \lambda) = \zeta_j^{\mathtt{rx}} \times \lambda. \quad (4)$$

For energy replenishment, $s_i$'s battery can be recharged only when $e_i < e_i^{\max}$. Moreover, the MC needs to move to $s_i$'s position to charge its battery. The MC's charging rate is a constant [37], as denoted by $\tau_{\mathtt{mc}}$ (in J/s).

### 3.2 Problem Description

The SR of a sensor $s_i \in \hat{\mathcal{S}}$ depends on its residual energy:

$$r_i = \begin{cases} r_i^{\mathtt{H}} & \text{if } e_i \geq \delta \times e_i^{\max} \\ r_i^{\mathtt{L}} & \text{if } e_i^{\min} \leq e_i < \delta \times e_i^{\max} \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where $r_i^{\mathtt{H}} > r_i^{\mathtt{L}}$ (measured in packets/s) and $e_i^{\min}/e_i^{\max} < \delta < 1$. The MC initially locates in a certain position close to the sink, known as its *point of departure (POD)*. In each round, the sink runs the proposed algorithm and notifies the MC of the execution result. Starting from the POD, the MC moves to visit sensors based on the execution result and then goes back to the POD. To make sure that the MC can complete the task in a round, its energy is set as follows:

$$e_{\mathtt{mc}} \geq \tilde{E}_{\mathtt{mv}} \times L(\mathcal{P}_{\mathtt{SH}}) + \sum\nolimits_{\forall s_i \in \hat{\mathcal{S}}} e_i^{\max}, \quad (6)$$

where $\tilde{E}_{\mathtt{mv}}$ is the amount of energy taken by the MC to move a unit distance, and $L(\mathcal{P}_{\mathtt{SH}})$ is the length of the shortest path $\mathcal{P}_{\mathtt{SH}}$ to visit all sensors and return to the POD. In Eq. (6), we assume that the MC visits every sensor and charges the sensor's battery from empty to full [38], so it can have enough energy in the worst case. Besides, when the MC goes back to the POD, its battery is replaced with a new one. Thus, the MC can fast perform the task in the next round.

**Algorithm 1:** The EPCS Algorithm

**Input:** Set $\hat{\mathcal{S}}$ of sensors and the POD $v_0$
**Output:** MC's moving path $\mathcal{P}$ and the charging time for each sensor on $\mathcal{P}$
1 Find the shortest path $\mathcal{P}_{\mathtt{SH}}$ to visit sensors in $\hat{\mathcal{S}}$ and return to $v_0$;
2 $\mathcal{P} \leftarrow \mathcal{P}_{\mathtt{SH}}$;
3 Estimate the average EER $\mu_{\mathtt{avg}}$ of workable sensors;
4 **foreach** $v_j \in \mathcal{P} \setminus \{v_0\}$ **do**
5  **if** SNC($v_j$) = *true* **then**
6  Remove $v_j$ from $\mathcal{P}$;
7  **else**
8  Use the FBC module to calculate the charging time for the sensor whose identification is $v_j$;

Then, the P&C problem asks how to decide the MC's moving path to visit some sensors in $\hat{\mathcal{S}}$ and the time to charge each visited sensor in each round, such that both survivability and throughput of sensors is maximized. Here, survivability is defined by the percentage of *workable sensors* in $\hat{\mathcal{S}}$, where a sensor $s_i$ is viewed as a workable sensor if the condition $e_i \geq e_i^{\min}$ holds. Throughput is estimated by the amount of sensing data sent to the sink. Table 1 summarizes notations.

## 4 The Proposed EPCS Algorithm

Algo. 1 shows EPCS's pseudocode. Given the POD (denoted by $v_0$), line 1 finds a shortest path $\mathcal{P}_{\mathtt{SH}} = \{v_0, v_1, \cdots, v_m, v_0\}$ to visit each sensor in $\hat{\mathcal{S}}$ and go back to $v_0$, where each $v_j \in \mathcal{P}$ ($j \neq 0$) denotes the identification of a sensor. This can be done by using a TSP approximation algorithm (below, we call it a *TSP method*) [39]. Then, path $\mathcal{P}$ is set to $\mathcal{P}_{\mathtt{SH}}$ by line 2.

Let $\hat{\mathcal{S}}_{\mathtt{A}} \subseteq \hat{\mathcal{S}}$ be the subset of workable sensors. If $\hat{\mathcal{S}}_{\mathtt{A}}$ is not empty, we then estimate the average *energy expenditure rate (EER)* of workable sensors on generating and sending sensing data in the next $t_{\mathtt{obs}}$ seconds as follows:

$$\mu_{\mathtt{avg}} = \frac{\sum_{s_i \in \hat{\mathcal{S}}_{\mathtt{A}}} \tilde{E}_{\mathtt{se}}(s_i, \Lambda_i) + \tilde{E}_{\mathtt{tx}}(s_i, s_j, \Lambda_i)}{|\hat{\mathcal{S}}_{\mathtt{A}}| \times t_{\mathtt{obs}}}, \quad (7)$$

$$\Lambda_i = r_i \times t_{\mathtt{obs}} \times \lambda, \quad (8)$$

where $\lambda$ is the average number of bits in a packet of sensing data, and $s_j$ is the receiver of $s_i$'s data. EER is used to decide the MC's charging strategy, as discussed in Section 4.2. Note that $\mu_{\mathtt{avg}}$ is a *predicted* value. In other words, EER in Eq. (7) can be fast calculated, instead of waiting $t_{\mathtt{obs}}$ seconds. Thus, the time lag (caused by the calculation of EER) between the timing to execute the EPCS algorithm and the timing to charge sensors can be almost neglected.

Except for $v_0$ (i.e., POD), some nodes may be removed from $\mathcal{P}$ to let the MC serve the sensors in need of energy as soon as possible. The code is shown in lines 4–8. For each node $v_j$ on $\mathcal{P}$, we employ the *skippable node checking (SNC)* module in Section 4.1 to check if it can be removed. If $v_j$ is not skippable, we adopt the *floating battery charging (FBC)* module in Section 4.2 to decide its charging time. The above operation is repeated until every node on $\mathcal{P}$ is checked.

**Algorithm 2:** The SNC Module

**Input:** Next node $v_j$ to be visited on $\mathcal{P}$
**Output:** True or false
1 **if** $v_j \neq v_m$ **then**
2  **if** $e_{v_j} > \Delta_{\mathtt{H}} \times e_{v_j}^{\max}$ *and* $e_{v_{j+1}} < \Delta_{\mathtt{L}} \times e_{v_{j+1}}^{\max}$ **then**
3  Return true;
4 **else**
5  **if** $e_{v_m} > \Delta_{\mathtt{H}} \times e_{v_m}^{\max}$ *and* $v_1$ *is skipped and* $e_{v_1} < \Delta_{\mathtt{H}} \times e_{v_1}^{\max} + \sigma$ **then**
6  Return true;
7 Return false;

### 4.1 Skippable Node Checking (SNC) Module

The TSP method finds the shortest path $\mathcal{P}_{\mathtt{SH}}$ to visit all sensors in $\hat{\mathcal{S}}$, but it does not guarantee that the MC can charge each sensor in time. Some sensors may use up energy soon, and their visiting sequences on $\mathcal{P}_{\mathtt{SH}}$ are arranged after other sensors with enough energy. After exhausting energy, such sensors will cease functioning, until the MC charges them. In this case, the survivability and throughput of sensors may reduce if the MC chooses $\mathcal{P}_{\mathtt{SH}}$ to be its path.

Thus, the SNC module "prunes" path $\mathcal{P}_{\mathtt{SH}}$ by removing some nodes with enough energy, so the MC can fast move to charge the sensors about to running out of energy. Given the next node $v_j$ to be visited on $\mathcal{P}$ (which is initially set to $\mathcal{P}_{\mathtt{SH}}$ by line 2 in Algo. 1), Algo. 2 shows SNC's pseudocode. If $v_j$ is not the last node on $\mathcal{P}_{\mathtt{SH}}$ (i.e., $v_m$), SNC uses two conditions (in line 2) to determine whether $v_j$ can be skipped. First, $v_j$ has enough energy (i.e., $e_{v_j} > \Delta_{\mathtt{H}} \times e_{v_j}^{\max}$). Thus, even if the MC does not charge $v_j$, $v_j$ can still have energy for operation. Second, $v_{j+1}$ (i.e., the node whose sequence is immediately after $v_j$) has little energy (i.e., $e_{v_{j+1}} < \Delta_{\mathtt{L}} \times e_{v_{j+1}}^{\max}$). In this case, the MC should replenish energy for $v_{j+1}$ as soon as possible. When both conditions are satisfied, the SNC module returns true and $v_j$ will be removed from $\mathcal{P}$. Here, $\Delta_{\mathtt{H}}$ and $\Delta_{\mathtt{L}}$ are percentages, where $\min_{\forall s_i \in \hat{\mathcal{S}}} \{e_i^{\min}/e_i^{\max}\} < \Delta_{\mathtt{L}} < \Delta_{\mathtt{H}} < 1$.

If the node to be visited is $v_m$ (i.e., the last node on $\mathcal{P}_{\mathtt{SH}}$), we check if $v_m$ can be skipped based on three conditions: (1) $e_{v_m} > \Delta_{\mathtt{H}} \times e_{v_m}^{\max}$, (2) $v_1$ has been skipped in this round, and (3) $e_{v_1} < \Delta_{\mathtt{H}} \times e_{v_1}^{\max} + \sigma$, where $\sigma$ is a small value. Conditions (2) and (3) imply that the first node $v_1$ on $\mathcal{P}_{\mathtt{SH}}$ is skipped by SNC but $v_1$'s energy is only slightly above the threshold (i.e., $\Delta_{\mathtt{H}} \times e_{v_1}^{\max}$). To compensate $v_1$ (for the fairness concern), $v_m$ will be skipped if condition (1) also holds, so the MC can quickly move to charge $v_1$ in the next round.

When SNC removes a node $v_j$, the segment "$v_{j-1} \rightarrow v_j \rightarrow v_{j+1}$" in $\mathcal{P}$ is replaced by a segment "$v_{j-1} \rightarrow v_{j+1}$", instead of recalculating the shortest path to visit the residual sensors by the TSP method again. With the triangle inequality, we derive that $\tilde{D}(v_{j-1}, v_{j+1}) < \tilde{D}(v_{j-1}, v_j) + \tilde{D}(v_j, v_{j+1})$. Thus, $\mathcal{P}$ is shorter than $\mathcal{P}_{\mathtt{SH}}$, so the MC can visit those sensors in urgent need of energy earlier and charge them faster. Theorem 1 gives the time complexity of recreating an alternative path (i.e., $\mathcal{P}$) by the SNC module.

***Theorem 1.*** Recreating an alternative path $\mathcal{P}$ (from $\mathcal{P}_{\mathtt{SH}}$) by the SNC module takes $O(n)$ time, where $|\hat{\mathcal{S}}| = n$.

*Proof:* $\mathcal{P}$ is initially set to $\mathcal{P}_{\mathtt{SH}}$, so it contains all sensors in $\hat{\mathcal{S}}$ and the POD $v_0$. Except for $v_0$, for each node $v_j$ on $\mathcal{P}$,

**Algorithm 3:** The FBC Module

**Input:** Sensor $s_i$ to be charged
**Output:** Charging time for $s_i$
1 **if** $|\hat{\mathcal{S}}_{\mathtt{A}}| > \gamma|\hat{\mathcal{S}}|$ *and* $\sum_{\forall s_j \in \hat{\mathcal{S}}_{\mathtt{A}}} e_j/|\hat{\mathcal{S}}_{\mathtt{A}}| > e_{\mathtt{th}}$ *and* $\mu_{\mathtt{avg}} < \mu_{\mathtt{th}}$ **then**
2  Calculate $e_i^{\mathtt{CH}}$ by the fully charging strategy;
3 **else**
4  Calculate $e_i^{\mathtt{CH}}$ by the partially charging strategy;
5 Return $e_i^{\mathtt{CH}}/\tau_{\mathtt{mc}}$;

SNC checks only the energy of that node and the following node (i.e., $e_{v_j}$ and $e_{v_{j+1}}$). Thus, it takes $O(1)$ time to do a check. Since there are at most $n$ sensors on $\mathcal{P}$, the time for SNC to check the whole path is $n \times O(1) = O(n)$. Moreover, when SNC decides to remove a node $v_j$, it replaces segment "$v_{j-1} \rightarrow v_j \rightarrow v_{j+1}$" with segment "$v_{j-1} \rightarrow v_{j+1}$" in $\mathcal{P}$. Evidently, doing the segment replacement requires merely a constant time. Since no more than $n$ nodes can be removed from $\mathcal{P}$, it takes $O(n)$ time to replace segments in the worst case. Therefore, the time complexity of the SNC module to recreate path $\mathcal{P}$ from $\mathcal{P}_{\mathtt{SH}}$ is $O(n) + O(n) = O(n)$. □

### 4.2 Floating Battery Charging (FBC) Module

Owing to adaptive SRs, sensors generate different amount of sensing data and have different speeds of energy consumption. Thus, the FBC module takes a *floating* charging policy, and Algo. 3 gives its pseudocode. Let $e_i^{\mathtt{CH}}$ be the amount of energy that the MC will charge a sensor $s_i$ on $\mathcal{P}$. Depending on the network status, FBC has two charging strategies.

*Fully charging strategy:* FBC adopts this strategy when three conditions obtain: (1) $|\hat{\mathcal{S}}_{\mathtt{A}}| > \gamma|\hat{\mathcal{S}}|$, where $0.8 \leq \gamma < 1$, (2) $\sum_{\forall s_j \in \hat{\mathcal{S}}_{\mathtt{A}}} e_j/|\hat{\mathcal{S}}_{\mathtt{A}}| > e_{\mathtt{th}}$, where $e_{\mathtt{th}}$ is an energy threshold, and (3) $\mu_{\mathtt{avg}} < \mu_{\mathtt{th}}$, where $\mu_{\mathtt{th}}$ is a threshold on EER. Here, conditions (1) and (2) mean that more than 80% of sensors in the WSN are workable, and most of them are energy-rich. Condition (3) implies that these sensors consume energy slowly. In this case, the MC can take a relatively short time to fully charge each sensor. Thus, we set $e_i^{\mathtt{CH}}$ by

$$e_i^{\mathtt{CH}} = e_i^{\max} - e_i. \tag{9}$$

*Partially charging strategy:* If any condition violates, the MC takes much time to charge sensors, as many sensors need to be charged, or they have little energy. To avoid some low-energy sensors waiting too long, the MC dynamically adjusts the charging amount. If $e_i \geq (1-\alpha)e_i^{\max}$, $e_i^{\mathtt{CH}}$ is set by Eq. (9). Otherwise, we set $e_i^{\mathtt{CH}}$ as follows:

$$e_i^{\mathtt{CH}} = \alpha e_i^{\max} + \beta(|\hat{\mathcal{S}}_{\mathtt{A}}|/|\hat{\mathcal{S}}|)\varrho. \tag{10}$$

where $0 < \alpha, \beta < 1$ and $\varrho = (1-\alpha)e_i^{\max} - e_i$. In Eq. (10), sensor $s_i$ is given a *basic charging amount* $\alpha e_i^{\max}$. If $s_i$'s battery still has room to charge, the MC gives it an *extra charging amount* $\beta(|\hat{\mathcal{S}}_{\mathtt{A}}|/|\hat{\mathcal{S}}|)\varrho$, which depends on the ratio of workable sensors to total sensors. Here, when there are fewer dead sensors needed to be recharged (i.e., $|\hat{\mathcal{S}}_{\mathtt{A}}|/|\hat{\mathcal{S}}|$ is larger), the MC can give more extra charging amount to $s_i$ for improving its throughput (referring to the SR in Eq. (5)).

Since the MC's charging rate $\tau_{\mathtt{mc}}$ is fixed, the charging time for $s_i$ is $e_i^{\mathtt{CH}}/\tau_{\mathtt{mc}}$. Theorem 2 shows the correctness of the FBC module, and Theorem 3 analyzes its time complexity.

***Theorem 2.*** The MC will never overcharge a sensor's battery by using the FBC module.

*Proof:* For each sensor $s_i \in \hat{\mathcal{S}}$, $e_i^{\max}$ is the capacity of its battery and $e_i$ is the current residual energy, where $e_i \leq e_i^{\max}$. Evidently, the MC can charge up to an amount $(e_i^{\max} - e_i)$ of energy for $s_i$, so it will not overcharge $s_i$'s battery by using Eq. (9). In Eq. (10), since $\beta < 1$ and $\hat{\mathcal{S}}_{\mathtt{A}} \subseteq \hat{\mathcal{S}}$, we derive that

$$\begin{aligned}&\alpha e_i^{\max} + \beta(|\hat{\mathcal{S}}_{\mathtt{A}}|/|\hat{\mathcal{S}}|)((1-\alpha)e_i^{\max} - e_i)\\ &< \alpha e_i^{\max} + 1 \times 1 \times ((1-\alpha)e_i^{\max} - e_i) = e_i^{\max} - e_i.\end{aligned}$$

Therefore, the MC will not overcharge $s_i$'s battery by using Eq. (10), thereby proving this theorem. □

***Theorem 3.*** Given $n$ sensors in $\hat{\mathcal{S}}$, the worst-case time complexity of the FBC module is $O(n)$.

*Proof:* FBC chooses between the fully or partially charging strategies by checking three conditions, and the check is performed once. It takes $O(1)$ time to check conditions (1) and (3) and $O(n)$ time to check condition (2). Since FBC finds charging time for only the sensors on path $\mathcal{P}$, the worst case occurs when all sensors in $\hat{\mathcal{S}}$ are on $\mathcal{P}$. The calculation in Eqs. (9) and (10) requires $O(1)$ time. Thus, FBC has time complexity of $O(1) + O(n) + |\hat{\mathcal{S}}| \times O(1) = O(n)$. □

### 4.3 Discussion

Let us discuss the rationale of EPCS. It uses a TSP method to find the shortest path $\mathcal{P}_{\mathtt{SH}}$ to visit all sensors in $\hat{\mathcal{S}}$, and then recreates an alternative path $\mathcal{P}$ by removing some nodes from $\mathcal{P}_{\mathtt{SH}}$. Doing so has two benefits. First, when the network is small (i.e., $\hat{\mathcal{S}}$ has fewer sensors or the sensing field is small), $\mathcal{P}$ is almost the optimal moving path for the MC to charge sensors, as only few nodes would be removed from $\mathcal{P}_{\mathtt{SH}}$ to create $\mathcal{P}$. Second, by skipping some energy-rich nodes, the MC can charge the sensors thirsty for energy faster, which helps improve the survivability of sensors in a large WSN.

EPCS contains the SNC and FBC modules, whose designs consider adaptive SRs. As sensors have different SRs and their SRs can dynamically change, their energy expenditure would be of big difference. Thus, SNC differentiates between energy-rich sensors (with small SRs) and energy-poor sensors (with large SRs). If the visiting sequences of energy-poor sensors are behind those of energy-rich sensors on $\mathcal{P}_{\mathtt{SH}}$, SNC lets the MC skip these energy-rich sensors to charge energy-poor sensors as soon as possible. Moreover, FBC chooses the partially charging strategy if EER $\mu_{\mathtt{avg}}$ exceeds a threshold, where the calculation of EER in Eq. (7) consults sensors' SRs. Specifically, when more sensors have larger SRs, $\mu_{\mathtt{avg}}$ rises accordingly. Thus, the MC partially charges most sensors to increase their survivability. On the contrary, if most sensors have smaller SRs, $\mu_{\mathtt{avg}}$ reduces, so the MC can fully charge sensors. In this case, the SR $r_i$ of a sensor $s_i$ can be set to $r_i^{\mathtt{H}}$ by Eq. (5), thereby improving its throughput. The above designs let our EPCS algorithm be suitable for sensors with adaptive SRs.

Theorem 4 analyzes the time complexity of EPCS. Excluding the time used to find the shortest path $\mathcal{P}_{\mathtt{SH}}$ by a TSP method, the residual part of EPCS spends just $O(n)$ time. In fact, EPCS adopts the TSP method only in the first round. Then, we can remove line 1 from Algo. 1 in other rounds. Thus, EPCS incurs less overhead in computation. Theorem 5 proves that the MC can complete the charging task assigned by EPCS, subject to its energy budget $e_{\mathtt{mc}}$.

TABLE 2: Parameters used in the simulation.

| *Parameter* | *Value* |
|---|---|
| **Sensor-related parameters:** | |
| Communication range | 150 m |
| Packet length | 500 bytes |
| Battery | $e_i^{\min}$: 1 J, $e_i^{\max}$: 6480 J |
| Coefficients for energy expenditure defined in Section 3.1 | $u_i^{\mathrm{se}}$: 1.5 V, $c_i^{\mathrm{se}}$: 25 mA, $t_i^{\mathrm{se}}$: 0.25 ms<br>$\zeta_i^{\mathrm{tx}}$: 40 nJ/bit, $\zeta_i^{\mathrm{am}}$: 80 pJ/bit (per $m^2$)<br>$\zeta_j^{\mathrm{rx}}$: 40 nJ/bit |
| SR (unit: packets/s)<br># of sensors: $r_i^{\mathrm{H}} \rightarrow r_i^{\mathrm{L}}$<br>Threshold $\delta$: 0.2 | 34 sensors: 1/36 → 1/144<br>33 sensors: 1/48 → 1/192<br>33 sensors: 1/72 → 1/288 |
| **MC-related parameters:** | |
| Movement | Velocity: 3 m/s, energy expense ($\tilde{E}_{\mathrm{mv}}$): 4 J/m |
| Charging rate ($\tau_{\mathrm{mc}}$) | 5 J/s |

***Theorem 4.*** Given $n$ sensors in $\hat{\mathcal{S}}$, EPCS's time complexity is $f(n) + O(n)$, where $f(n)$ is the time required by a TSP method to find the shortest path $\mathcal{P}_{\mathrm{SH}}$.

*Proof:* In Algo. 1, line 1 spends $f(n)$ time to build $\mathcal{P}_{\mathrm{SH}}$. Line 3 finds EER $\mu_{\mathrm{avg}}$ by Eq. (7), which takes $O(n)$ time in the worst case (i.e., when $\hat{\mathcal{S}}_{\mathrm{A}} = \hat{\mathcal{S}}$). According to Theorems 1 and 3, the for-loop in lines 4–8 consumes $O(n)$ time. Thus, the time complexity of EPCS is $f(n) + O(n)$. □

***Theorem 5.*** The MC must have enough energy to complete the charging task assigned by EPCS in each round.

*Proof:* As discussed in Section 4.1, the path $\mathcal{P}$ recreated by the SNC module is no longer than the shortest path $\mathcal{P}_{\mathrm{SH}}$ found by the TSP method, so we obtain that $L(\mathcal{P}) \leq L(\mathcal{P}_{\mathrm{SH}})$. For each sensor $s_i$ on $\mathcal{P}$, the MC charges it with an amount $e_i^{\mathrm{CH}}$ of energy, where $e_i^{\mathrm{CH}} \leq e_i^{\max}$ by Theorem 2. Thus, the amount of energy required by the MC to complete its task is $\tilde{E}_{\mathrm{mv}} \times L(\mathcal{P}) + \sum_{\forall s_i \in \mathcal{P}} e_i^{\mathrm{CH}} \leq \tilde{E}_{\mathrm{mv}} \times L(\mathcal{P}_{\mathrm{SH}}) + \sum_{\forall s_i \in \hat{\mathcal{S}}} e_i^{\max}$, which is below its energy budget $e_{\mathrm{mc}}$ defined in Eq. (6). □

## 5 Performance Evaluation

We develop a simulator in C++ for performance evaluation, and Table 2 gives its parameters. The sensing field is a $K \times K$ square, where $K$ is set to 600, 800, and 1000 meters. There are 100 sensors randomly deployed in the sensing field. They route packets by using the *LEACH-C (low-energy adaptive clustering hierarchy–centralized) protocol* [40], which picks energy-rich sensors as cluster heads for the routing purpose. When a sensor acts as the cluster head, it consumes energy faster than other sensors [41]. Besides, we choose the *simulated annealing algorithm* [42] to be the TSP method (below, it is called "TSP-SA" for short). Except for TSP-SA, we compare our EPCS algorithm with the FCFS [28], NJNP [29], EDF [30], and TADP [31] methods discussed in Section 2.

Each *round* begins at the time when the MC departs from the POD and ends at the time as it goes back to the POD. Since different methods choose varied subsets of sensors in $\hat{\mathcal{S}}$ to be charged and find different moving paths for the MC to visit them, the period of a round will not be equal in different methods. Even if a method finds a fixed moving path for the MC (e.g., TSP-SA), its round period is not constant, because sensors have varied residual energy (and their charging time change accordingly) in different rounds. Thus, it is difficult and unfair to compare the survivability and throughput of sensors by different methods in a per-round fashion. Instead, we evaluate the system performance every 10,000 seconds, and the total simulation time is set to 1,000,000 seconds.

Our EPCS algorithm includes some parameters. Since the minimum required energy $e_i^{\min}$ for a sensor $s_i$ to operate is 1 J, we set $\Delta_{\mathrm{L}} = 0.1\%$ and $\Delta_{\mathrm{H}} = 3\%$ in the SNC module. In this case, $s_i$ will use up energy soon when $e_i < \Delta_{\mathrm{L}} \times e_i^{\max} = 0.1\% \times 6480\,\mathrm{J} = 6.48\,\mathrm{J}$. On the other hand, if $e_i > \Delta_{\mathrm{H}} \times e_i^{\max} = 3\% \times 6480\,\mathrm{J} = 194.4\,\mathrm{J}$, $s_i$ has enough energy for operation. For the FBC module, we set $\gamma$ = 0.85, $e_{\mathrm{th}}$ = 5500 J, and $\mu_{\mathrm{th}}$ = 0.6 J/s. In other words, when (1) more than 85% of sensors in $\hat{\mathcal{S}}$ are workable, (2) the workable sensors have at least $(5500/6480 \approx 85\%)$ of energy left, and (3) the average EER is below 0.6 J/s, the MC adopts the fully charging strategy. Otherwise, it employs the partially charging strategy. Here, when $s_i$'s EER is fixed to 0.6 J/s, it can keep operating for three hours. That is why we set $\mu_{\mathrm{th}}$ to 0.6 J/s. In Eq. (10), both $\alpha$ and $\beta$ are set to 0.2. In Section 5.4, we will investigate their effect on EPCS's performance.

### 5.1 Comparison on Survivability

First, we measure the survivability of sensors, as shown in Fig. 1, which is defined as the percentage of workable sensors (i.e., $\hat{\mathcal{S}}_{\mathrm{A}}$) in the WSN. Evidently, higher survivability implies that there exist more workable sensors to maintain the network operations (e.g., collecting data and routing packets). On the whole, the survivability in each method drops drastically in the first 100,000 seconds and then keeps stable (with some oscillations) afterward. This phenomenon points out the MC's limitation on charging sensors. Specifically, even though the MC has sufficient energy to charge every sensor in each round, it cannot keep all sensors workable due to some physical constraints (e.g., constant moving speed and charging rate). This also reveals why it is important to find an efficient solution for the P&C problem.

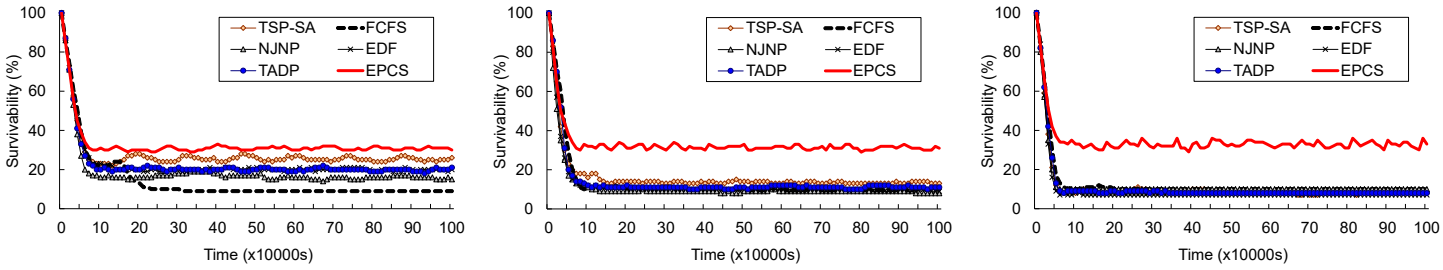
In Fig. 1(a), FCFS results in the lowest survivability, as it does not care about the MC's moving distance. By considering the residual energy of sensors, both EDF and TADP have higher survivability than NJNP. Since TSP-SA finds the shortest path $\mathcal{P}_{\mathrm{SH}}$ for the MC to visit sensors, its moving distance greatly reduces. In this case, the MC has more time to charge sensors, thereby raising their survivability. EPCS not only shorten $\mathcal{P}_{\mathrm{SH}}$ by removing some energy-rich nodes, but also lets the MC charge sensors in a floating fashion. Thus, our EPCS algorithm always has the highest survivability.

From Fig. 1(b) and (c), when the size of the sensing field increases (i.e., $K \geq 800$), except for EPCS, all other methods have very low survivability after 100,000 seconds, because the MC's moving distance substantially increases. Specifically, the survivability of TSP-SA, FCFS, NJNP, EDF, and TADP is below 14% and 10% after 100,000 seconds when $K$ is 800 and 1000, respectively. On the contrary, the survivability of our EPCS algorithm is always higher than 30%. This result shows the high efficiency of EPCS in terms of survivability, especially in a large sensing field.

### 5.2 Comparison on Throughput

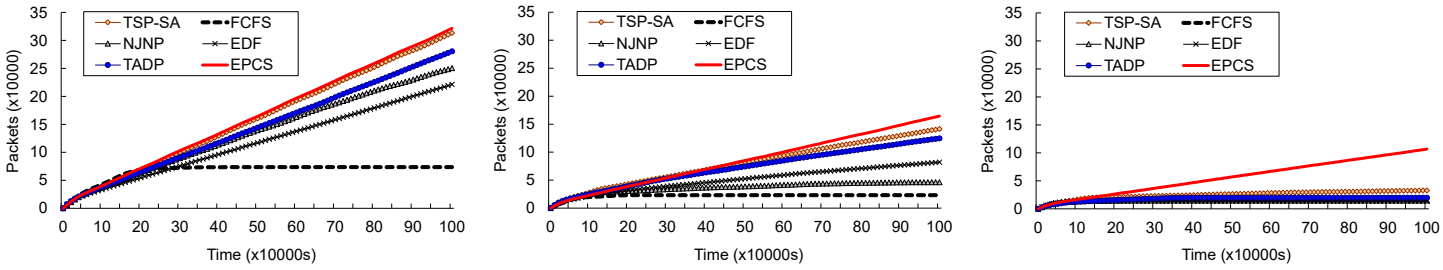
Next, we evaluate the *aggregate number of packets (ANP)* of sensing data successfully sent to the sink, as shown in Fig. 2. When there exist more workable sensors in the WSN, and most of them have sufficient energy to support high SRs $r_i^{\mathrm{H}}$ (i.e., $e_i \geq \delta \times e_i^{\max}$), they can send more sensing data to the sink. In this case, network throughput will rise accordingly.

In a small sensing field (i.e., $K = 600$), the MC can visit more sensors in $\hat{\mathcal{S}}$. Thus, the ANP of sensing data rises as




(a) $K = 600$ (b) $K = 800$ (c) $K = 1000$

Fig. 1: Comparison on the survivability of sensors by different methods.




(a) $K = 600$ (b) $K = 800$ (c) $K = 1000$

Fig. 2: Comparison on the ANP of sensing data sent to the sink by different methods.

TABLE 3: Improvement ratio of the ANP of sensing data by EPCS.

| $K$ | TSP-SA | FCFS | NJNP | EDF | TADP |
|---|---|---|---|---|---|
| 600 | 2.7% | 337.0% | 28.5% | 45.5% | 14.7% |
| 800 | 16.3% | 609.2% | 255.2% | 100.7% | 31.9% |
| 1000 | 225.4% | 571.3% | 623.4% | 680.7% | 440.9% |

time goes by. FCFS is the only exception, as it has very low survivability (i.e., most sensors are not workable). Both TSP-SA and EPCS find the shortest moving path for the MC to charge sensors, so they result in the largest ANP of sensing data. When $K$ is set to 800, the MC takes more time to visit sensors. Many sensors could not be charged in time, thereby reducing the ANP of sensing data. EPCS outperforms other methods, and the performance gap between EPCS and TSP-SA also increases, as shown in Fig. 2(b).

In a large sensing field (i.e., $K = 1000$), except EPCS, the ANP of sensing data in other methods stops growing after 400,000 seconds. The reason can be found in Fig. 1(c), where very few sensors are workable. Therefore, the network almost collapses, since the LEACH-C protocol cannot select enough sensors to be cluster heads. Thanks to both SNC and FBC modules, our EPCS algorithm keeps more than 30% of sensors workable and avoids this predicament. Table 3 summarizes the improvement ratio of the ANP of sensing data by EPCS as compared with other methods. As can be seen, our EPCS algorithm can significantly improve the ANP of sensing data, especially in a large sensing field.

Then, we measure the average throughput of workable sensors (i.e., $\hat{\mathcal{S}}_{\mathtt{A}}$), which can be viewed as the throughput of each communication in the WSN. To do so, we divide the simulation time into 100 periods, where the period length is set to 10,000 seconds. Periods 30, 60, and 90 are selected as the sampling periods, which fall within the early, middle, and late stages in the simulation, respectively. Fig. 3 gives the experimental result. As time goes by, there are fewer workable sensors and some of them may not be charged with enough energy. Thus, the average throughput reduces accordingly. In a small sensing field (i.e., $K = 600$), except for FCFS, all other methods have relatively high throughput. EPCS has the highest throughput, followed by TSP-SA, TADP, EDF, and NJNP. The similar phenomenon also happens when $K = 800$. On the other hand, in a large sensing field (i.e., $K = 1000$), only TSP-SA and EPCS still have non-zero throughput[1] in every sampling period, where EPCS has higher throughput than TSP-SA. This result shows the efficiency of our EPCS algorithm on improving sensors' throughput.

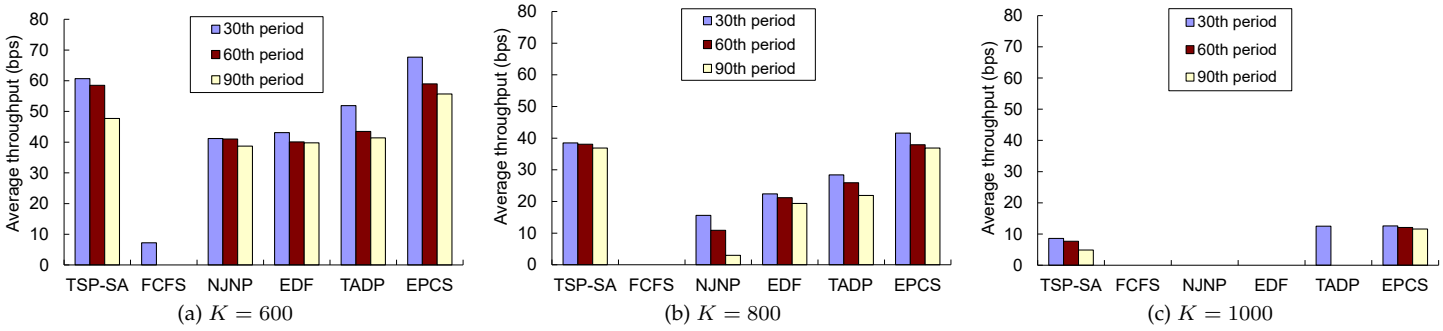
### 5.3 Energy Distribution and Connectivity of Sensors

Then, we study the energy distribution of sensors. To do so, we divide the workable sensors (i.e., $\hat{\mathcal{S}}_{\mathtt{A}}$) into seven groups based on their residual energy, as shown in Fig. 4, and measure the number of sensors in each group. As discussed in Sections 5.1 and 5.2, TSP-SA outperforms all other methods (except EPCS). Besides, the EPCS algorithm uses TSP-SA to find the shortest path $\mathcal{P}_{\mathtt{SH}}$. Thus, we compare the energy distribution of sensors only in TSP-SA and EPCS.

As discussed in Section 4.3, the alternative path $\mathcal{P}$ recreated by EPCS would be similar to the shortest path $\mathcal{P}_{\mathtt{SH}}$ found by TSP-SA in a small sensing field (i.e., $K = 600$). As the visiting sequences of sensors for the MC in both EPCS and TSP-SA are similar, they have similar energy distributions. In addition, some sensors can be fully charged, so they have more residual energy (i.e., $e_i > 4000\,\mathrm{J}$).

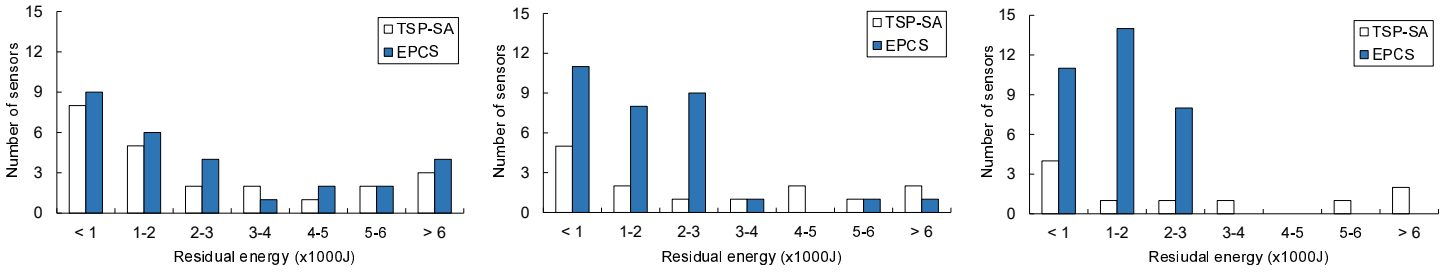
When $K \geq 800$, the MC spends more time on movement, so more sensors may exhaust their energy before the MC charges them. As TSP-SA takes the fully charging strategy, the energy distribution of sensors becomes unbalanced. Few sensors have nearly full energy (i.e., $e_i \geq 5000\,\mathrm{J}$), but many

1. In other methods, although there exist workable sensors, these sensors cannot form a connected network. In this case, they cannot send data to the sink, so the average throughput falls to zero.




(a) $K = 600$ (b) $K = 800$ (c) $K = 1000$

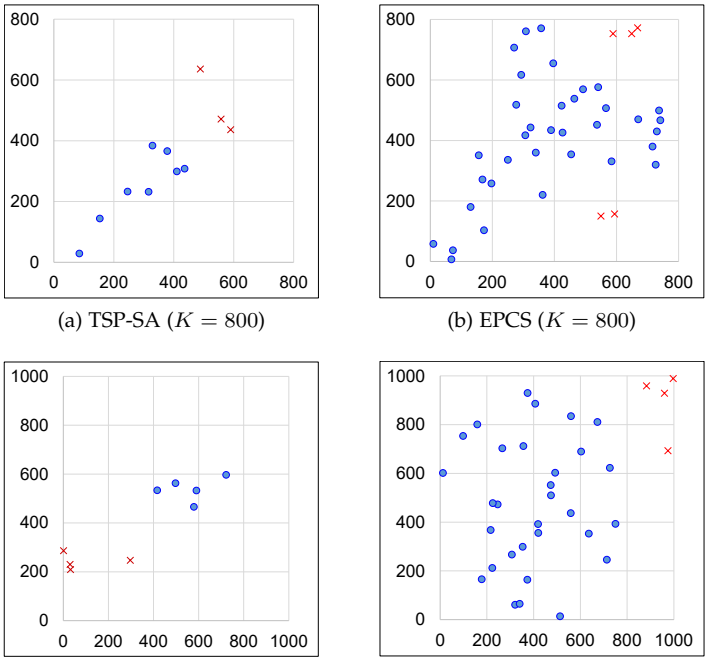
Fig. 3: Comparison on the average throughput of workable sensors by different methods.




(a) $K = 600$ (b) $K = 800$ (c) $K = 1000$

Fig. 4: Comparison on the energy distribution of sensors by TSP-SA and EPCS.

sensors cannot be charged. On the other hand, EPCS flexibly adjusts the amount of energy charged to sensors by Eq. (10). Thus, the MC can charge more sensors, each with energy enough to maintain its operations (i.e., $e_i < 3000$ J). In this way, EPCS can evenly distribute energy among sensors, thereby improving their survivability and throughput.



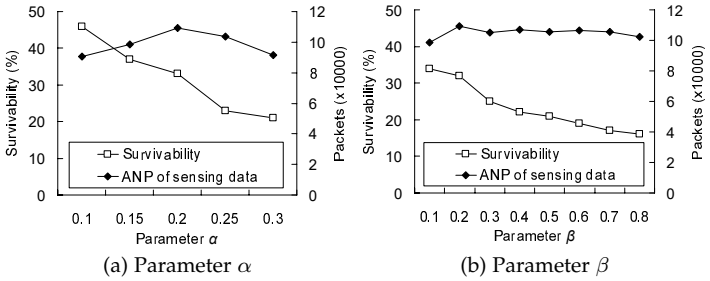
(a) TSP-SA ($K = 800$) (b) EPCS ($K = 800$)

(c) TSP-SA ($K = 1000$) (d) EPCS ($K = 1000$)

Fig. 5: Locations of workable sensors in TSP-SA and EPCS.

Afterward, we study the connectivity of the network in TSP-SA and EPCS. Fig. 5 shows the locations of workable sensors by setting $K$ to 800 and 1000, where the sink and POD are in the center of the sensing field. Dots (•) signify the sensors that can form a connected network to send data to the sink, while crosses (×) denote the sensors that are workable but isolated. As can be seen, EPCS makes many more sensors workable as compared with TSP-SA. That is because EPCS lets the MC charge each sensor with enough energy for operation, so as to charge more sensors and improve their survivability. Besides, EPCS skips only energy-rich sensors, so the network connectivity is still maintained. From Fig. 5, there are only 8 and 5 workable sensors forming connected networks in TSP-SA when $K$ is 800 and 1000, respectively. On the other hand, 35 and 31 workable sensors can be connected in EPCS when $K$ is 800 and 1000, respectively. This result shows the superiority of EPCS over TSP-SA.

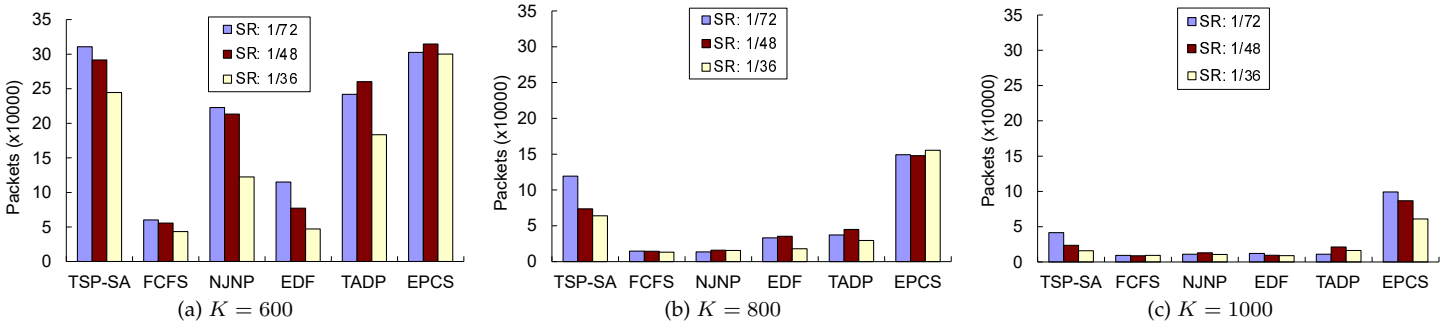
### 5.4 Effect of Parameters




(a) Parameter $\alpha$ (b) Parameter $\beta$

Fig. 6: Effect of parameters $\alpha$ and $\beta$ on EPCS's performance.

In the partially charging strategy, the MC finds the amount of energy charging to a sensor by Eq. (10), which is controlled by parameters $\alpha$ and $\beta$. In this section, we study their effect on EPCS's performance, where $K = 1000$. Fig. 6(a) shows




(a) $K = 600$ (b) $K = 800$ (c) $K = 1000$

Fig. 7: Comparison on the ANP of sensing data sent to the sink in the scenario of constant SRs.

$\alpha$'s effect, where $\beta = 0.2$. Since the basic charging amount is $\alpha e_i^{\max}$, changing the value of $\alpha$ has a great impact on the performance. The survivability reduces when $\alpha$ rises, since the MC spends more time on charging each visited sensor. Besides, some sensors are charged with more energy, so they can generate more data (based on Eq. (5)). Thus, the ANP of sensing data increases by raising $\alpha$. However, when $\alpha$ is larger than 0.2, the ANP of sensing data decreases instead due to low survivability of sensors. From Fig. 6(a), a good value for $\alpha$ is 0.2, as we can keep higher survivability and ANP of sensing data.

Fig. 6(b) shows the effect of $\beta$, which decides the ratio of extra charging amount, where $\alpha = 0.2$. Since each visited sensor is given a basic charging amount, $\beta$'s effect is less than $\alpha$'s effect. That is why the trend in Fig. 6(b) resembles the trend in Fig. 6(a), but the scale of the x-axis in Fig. 6(b) (i.e., $\beta = 0.1 \sim 0.8$) is larger than that in Fig. 6(a) (i.e., $\alpha = 0.1 \sim 0.3$). Based on Fig. 6(b), we suggest setting $\beta$ to 0.2 to get high survivability and ANP of sensing data.

### 5.5 Scenario of Constant SRs

Afterward, let us consider the scenario where sensors have the same and constant SRs. Specifically, the SRs of all sensors are fixed to 1/72, 1/48, and 1/36 packets/s, and we evaluate the ANP of sensing data sent to the sink after 1,000,000 seconds. Fig. 7 shows the experimental result. Since sensors will not adjust SRs by Eq. (5), their energy expenditure may keep constant, even if they have less residual energy. In this case, sensors take a short time to use up energy, especially when the SR is set to 1/36 packets/s. In fact, sensors have much larger EERs in this scenario, as compared with the original scenario of adaptive SRs in Table 2. That is why the ANP of sensing data in Fig. 7 is lower than that in Fig. 2.

As discussed in Section 5.2, the ANP of sensing data reduces if $K$ increases. When the SR rises, sensors generate packets of sensing data more frequently, which increases the ANP of sensing data. On the other hand, since sensors have large EERs, their survivability decreases accordingly. Therefore, there will exist fewer workable sensors to generate packets, thereby decreasing the ANP of sensing data. Based on the above two reasons, the ANP of sensing data in each method may not necessarily increase (or decrease) as the SR increases. As compared with TSP-SA, FCFS, NJNP, EDF, and TADP, our EPCS algorithm always results in the highest ANP of sensing data, which demonstrates that EPCS also has the best performance among all methods when sensors have constant SRs and large EERs.

## 6 Conclusion and Future Work

In many WSN applications, sensors equipped with small batteries have to work for a long time. This paper uses an MC to charge sensors with adaptive SRs, and proposes the EPCS algorithm to arrange the MC's path and charging time to improve the survivability and throughput of sensors. With the SNC module, the MC flexibly skips some energy-rich sensors on the shortest path found by the TSP method, thereby charging those sensors in urgent need of energy as soon as possible. Besides, the FBC module offers a floating charging mechanism based on the network status. Simulation results show that our EPCS algorithm not only keeps more sensors workable but also increases their throughput, as compared with the TSP-SA, FCFS, NJNP, EDF, and TADP methods. Moreover, EPCS can evenly distribute energy among sensors and maintain network connectivity.

Finally, we give some future directions. First, the SNC module provides a low-complexity way to recreate an alternative path from $\mathcal{P}_{\mathtt{SH}}$, but the path may not necessarily be optimal. As future work, we expect to develop a more sophisticated method to improve the SNC module. Second, it deserves further investigation on how to well dispatch MCs to extend the lifetime of a WSN composed of multi-attribute sensors [43] and heterogeneous sensors [44]. In this case, we have to consider multiple factors when selecting sensors to be charged, including the residual energy, position, importance, and capability of each sensor. Third, some protocols dedicated to IoT devices, such as the *constrained application protocol (CoAP)*, affect both sensing and reporting rates of sensors [45]. In view of this, we will consider developing efficient scheduling algorithms for MCs to charge sensors that adopt these protocols.

## References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: automation networks in the era of the Internet of things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[2] Y.C. Wang, F.J. Wu, and Y.C. Tseng, "Mobility management algorithms and applications for mobile sensor networks," *Wireless Comm. and Mobile Computing*, vol. 12, no. 1, pp. 7–21, 2012.

[3] Y.C. Wang and G.W. Chen, "Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks," *IEEE Trans. Vehicular Technology*, vol. 66, no. 8, pp. 7234–7248, 2017.

[4] A. Alaiad and L. Zhou, "Patients' adoption of WSN-based smart home healthcare systems: an integrated model of facilitators and barriers," *IEEE Trans. Professional Comm.*, vol. 60, no. 1, pp. 4–23, 2017.

[5] Y.C. Wang and W.T. Chen, "An automatic and adaptive light control system by integrating wireless sensors and brain-computer interface," *Proc. IEEE Int'l Conf. Applied System Innovation*, 2017, pp. 1399–1402.

[6] Y.C. Wang and S.E. Hsu, "Deploying R&D sensors to monitor heterogeneous objects and accomplish temporal coverage," *Pervasive and Mobile Computing*, vol. 21, pp. 30–46, 2015.
[7] M.E. Bayrakdar, "A smart insect pest detection technique with qualified underground wireless sensor nodes for precision agriculture," *IEEE Sensors J.*, vol. 19, no. 22, pp. 10892–10897, 2019.
[8] Y.C. Wang and C.C. Yang, "3S-cart: a lightweight, interactive sensor-based cart for smart shopping in supermarkets," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6774–6781, 2016.
[9] N.A. Pantazis, S.A. Nikolidakis, and D.D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Comm. Surveys & Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.
[10] Y.C. Wang, "Data compression techniques in wireless sensor networks," in *Pervasive Computing*, Hauppauge: Nova Science Publishers, 2012.
[11] R. Elhabyan, W. Shi, and M. St-Hilaire, "Coverage protocols for wireless sensor networks: review and future directions," *J. Comm. and Networks*, vol. 21, no. 1, pp. 45–60, 2019.
[12] D. Ma, G. Lan, M. Hassan, W. Hu, and S.K. Das, "Sensing, computing, and communications for energy harvesting IoTs: a survey," *IEEE Comm. Surveys & Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2020.
[13] X. Lu, P. Wang, D. Niyato, D.I. Kim, and Z. Han, "Wireless charging technologies: fundamentals, standards, and network applications," *IEEE Comm. Surveys & Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2016.
[14] Y.C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, vol. 47, no. 1, pp. 12:1–12:36, 2014.
[15] Y.C. Wang and C.T. Wei, "Lightweight, latency-aware routing for data compression in wireless sensor networks with heterogeneous traffics," *Wireless Comm. and Mobile Computing*, vol. 16, no. 9, pp. 1035–1049, 2016.
[16] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 14, no. 9, pp. 1861–1875, 2015.
[17] D. Arivudainambi and S. Balaji, "Optimal placement of wireless chargers in rechargeable sensor networks," *IEEE Sensors J.*, vol. 18, no. 10, pp. 4212–4222, 2018.
[18] 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 Trans. Mobile Computing*, vol. 19, no. 8, pp. 1910–1927, 2020.
[19] H. Dai, Y. Liu, N. Yu, C. Wu, G. Chen, T. He, and A.X. Liu, "Radiation constrained wireless charger placement," *IEEE/ACM Trans. Networking*, vol. 29, no. 1, pp. 48–64, 2021.
[20] N. Yu, H. Dai, G. Chen, A.X. Liu, B. Tian, and T. He, "Connectivity-constrained placement of wireless chargers," *IEEE Trans. Mobile Computing*, vol. 20, no. 3, pp. 909–927, 2021.
[21] P. Guo, X. Liu, S. Tang, and J. Cao, "Concurrently wireless charging sensor networks with efficient scheduling," *IEEE Trans. Mobile Computing*, vol. 16, no. 9, pp. 2450–2463, 2017.
[22] Y.C. Wang and J.W. Huang, "Efficient dispatch of mobile sensors in a WSN with wireless chargers," *Pervasive and Mobile Computing*, vol. 51, pp. 104–120, 2018.
[23] 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 Trans. Networking*, vol. 25, no. 5, pp. 3161–3174, 2017.
[24] L. Fu, L. He, P. Cheng, Y. Gu, J. Pan, and J. Chen, "ESync: energy synchronized mobile charging in rechargeable wireless sensor networks," *IEEE Trans. Vehicular Technology*, vol. 65, no. 9, pp. 7415–7431, 2016.
[25] F. Chen, Z. Zhao, G. Min, and Y. Wu, "A novel approach for path plan of mobile chargers in wireless rechargeable sensor networks," *Proc. IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Networks*, 2016, pp. 63–68.
[26] N. Wang, J. Wu, and H. Dai, "Bundle charging: wireless charging energy minimization in dense wireless sensor networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, 2019, pp. 810–820.
[27] J. Liu and F. Feng, "Optimization and control of wireless rechargeable sensor network based on intelligent algorithm," *Proc. World Conf. Mechanical Engineering and Intelligent Manufacturing*, 2020, pp. 126–130.
[28] L. He, Z. Yang, J. Pan, L. Cai, J. Xu, and Y. Gu, "Evaluating service disciplines for on-demand mobile data collection in sensor networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 4, pp. 1861–1875, 2014.
[29] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 14, no. 9, pp. 1861–1875, 2015.
[30] W. Xu, W. Liang, X. Jia, and Z. Xu, "Maximizing sensor lifetime in a rechargeable sensor network via partial energy charging on sensors," *Proc. IEEE Int'l Conf. Sensing, Comm., and Networking*, 2016, pp. 1–9.
[31] C. Lin, Z. Wang, D. Han, Y. Wu, C.W. Wu, and G. Wu, "TADP: Enabling temporal and distantial priority scheduling for on-demand charging architecture in wireless rechargeable sensor networks," *J. Systems Architecture*, vol. 70, pp. 26–38, 2016.
[32] S.A. Chowdhury, A. Benslimane, and F. Akhter, "Autonomous mobile chargers for rechargeable sensor networks using space filling curve," *Proc. IEEE Int'l Conf. Comm.*, 2018, pp. 1–6.
[33] R.M. Buehrer, H. Wymeersch, and R.M. Vaghefi, "Collaborative sensor network localization: algorithms and practical issues," *Proc. the IEEE*, vol. 106, no. 6, pp. 1089–1114, 2018.
[34] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.
[35] G.E. Jan, K. Fung, C. Luo, and C.C. Kuan, "Planning a shortest path based on detour and path shortening," *Proc. IEEE Southeastcon Conf.*, 2019, pp. 1–4.
[36] Y.C. Wang and K.C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Trans. Mobile Computing*, vol. 18, no. 7, pp. 1527–1540, 2019.
[37] X. Lu, D. Niyato, P. Wang, D.I. Kim, and Z. Han, "Wireless charger networking for mobile devices: Fundamentals, standards, and applications," *IEEE Wireless Comm.*, vol. 22, no. 2, pp. 126–135, 2015.
[38] Y.C. Wang, W.C. Peng, and Y.C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1836–1850, 2010.
[39] D.P. Williamson and D.B. Shmoys, *The Design of Approximation Algorithms*, Cambridge: Cambridge University Press, 2010.
[40] S.K. Singh, P. Kumar, and J.P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, vol. 5, pp. 4298–4328, 2017.
[41] Y.C. Wang and S.W. Yeh, "E-DSR: energy-efficient routing for sensors with diverse sensing rates," *Int'l J. Ad Hoc and Ubiquitous Computing*, vol. 34, no. 4, pp. 233–248, 2020.
[42] H. Bayram and R. Sahin, "A new simulated annealing approach for travelling salesman problem," *Mathematical and Computational Applications*, vol. 18, no. 3, pp. 313–322, 2013.
[43] Y.C. Wang, "A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network," *IEEE Trans. Mobile Computing*, vol. 13, no. 4, pp. 709–722, 2014.
[44] S.K. Roy, S. Misra, and N.S. Raghuwanshi, "SensPnP: seamless integration of heterogeneous sensors with IoT devices," *IEEE Trans. Consumer Electronics*, vol. 65, no. 2, pp. 205–214, 2019.
[45] W.K. Lai, Y.C. Wang, and S.Y. Lin, "Efficient scheduling, caching, and merging of notifications to save message costs in IoT networks using CoAP," *IEEE Internet of Things J.*, vol. 8, no. 2, pp. 1016–1029, 2021.