Cooperative Wireless Charging Vehicle Scheduling

Huanyang Zheng and Jie Wu
Center for Networked Computing, Temple University, USA
Email: {huanyang.zheng, jiewu}@temple.edu

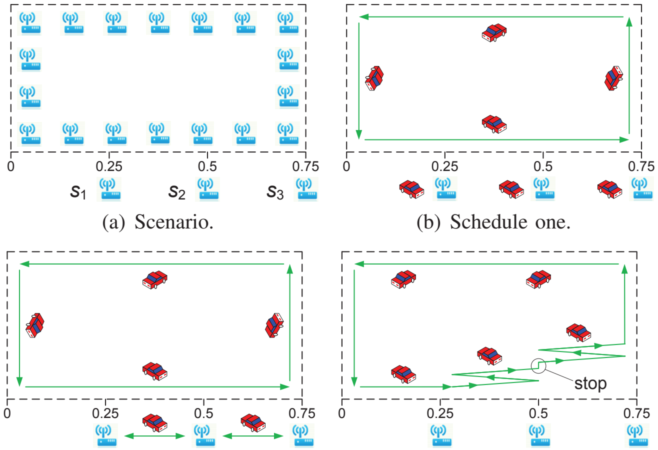
***Abstract*—Recent breakthroughs in wireless energy transfer-based rechargeable batteries enable a promising application of Wireless Charging Vehicles (WCVs) in Wireless Rechargeable Sensor Networks (WRSNs). This paper studies cooperative WCV schedules in WRSNs to optimize sensor recharging. The objective is to minimize the number of WCVs under the constraint that all sensors must be periodically recharged before running out of energy (i.e., before lifetime). Our problem is NP-hard and is very challenging due to the complexity of WCV route schedules. WCVs can be used to recharge sensors in turn. Our problem is thoroughly explored in line, cycle, and metric spaces (such as a three-dimensional Euclidean space). In terms of line and cycle spaces, greedy algorithms with ratios of 2 and 4, respectively, are proposed. By exploring two WCV schedule patterns, the optimal algorithm is found for the cycle space when sensor lifetimes are identical. For the metric space with an identical sensor lifetime, an algorithm with a ratio of 2.5 is proposed through constructing the minimum distance forest among sensors. It is also extended to the metric space with non-identical sensor lifetimes by grouping sensors according to their lifetimes. Finally, real data-driven experiments demonstrate the efficiency and effectiveness of the proposed approximation algorithms.**

***Keywords*-Wireless charging vehicles, cooperative schedule, wireless rechargeable sensor networks, approximation.**

I. Introduction

The limited battery capacities of sensors became the biggest problem in the applicability of Wireless Rechargeable Sensor Networks (WRSNs) over the past decade. Although research efforts have been made to prolong the sensors' lifetimes [1, 2], battery capacity remains a performance bottleneck that hinders the deployment of WRSNs. Recent breakthroughs in wireless power transfers enable a promising application of Wireless Charging Vehicles (WCVs) in WRSNs. WCVs can serve as mobile chargers that recharge the sensors through the wireless power transfer technology, which has been adopted in WiFi and RFID sensing devices [3]. The literature [4, 5] demonstrates that significant energy and cost savings, as well as extended life spans of WRSNs, can be achieved by using WCVs to periodically recharge the sensors.

In the existing literatures [4, 6], WCVs are independently scheduled to recharge sensors in WRSNs. In contrast, this paper studies cooperative WCV schedules to improve the efficiency. In our scenario, sensors are placed in a metric space (e.g., a three-dimensional Euclidean space), and the distances between them are known a priori. The lifetime of each sensor is known. Each sensor needs to be recharged before running out of energy. WCVs with identical speeds are used to periodically charge sensors. The range of the wireless power transfer is negligible with respect to the distance between the sensors. According to [7], the wireless power transfer is most efficient within a range of half a meter. Meanwhile, the distance between sensors can be hundreds of meters in large-scale WRSNs [8]. Hence, WCVs should reach the sensors before recharging them. Since WCVs have limited speeds, we need multiple WCVs to accomplish the sensor recharging task.



(a) Scenario.

(b) Schedule one.

(c) Schedule two.

(d) Optimal Schedule.

Fig. 1. Toy example to illustrate the problem complexity.

The cost of a sensor recharging schedule depends on its WCVs. To save the operation cost, the objective of this paper is to *minimize the number of WCVs* under the constraint that sensors must be periodically recharged before running out of energy (i.e., sensors are running forever). Our problem is challenging due to the complexity of WCV route schedules. A toy example is shown in Fig. 1(a). It includes a 1.125km×0.75km rectangle, whose length is compressed to save space. The rectangle is densely covered by sensors that have lifetimes of 1h without recharging. Three additional special sensors (s_1 at 0.25m, s_2 at 0.5m, and s_3 at 0.75m) are deployed on an edge of the rectangle. Lifetimes of s_1, s_2, and s_3 are 0.5h, 0.25h, and 0.5h, respectively. The speed of the WCVs is 1km/h. Fig. 1(b) illustrates a naive schedule, which uses 7 WCVs to guarantee that sensors are periodically recharged before running out of energy. It uses 4 WCVs to evenly traverse the rectangle. Since the rectangle circumference is 3.75km, all sensors except s_1, s_2, and s_3 can be periodically recharged before running out of energy. The naive schedule also uses 3 WCVs to recharge s_1, s_2, and s_3. Fig. 1(c) shows a better schedule, which uses 6 WCVs in total and uses only 2 WCVs to recharge s_1, s_2, and s_3. The 1st WCV goes back and forth between s_1 and s_2, and the 2nd WCV goes back and forth between s_2 and

s_3. Meanwhile, the 1st WCV arrives at s_2 0.25h earlier than the 2nd WCV (they do not arrive at s_2 at the same time). However, the optimal schedule is more interesting, as shown in Fig. 1(d). It uses only 5 WCVs to evenly traverse the rectangle, however, a magic schedule happens when a WCV arrives at s_1. Supposing that each operation takes 0.25h, each WCV performs the following 7 operations in sequence after arriving s_1: (1) go to s_2, (2) go back to s_1, (3) go to s_2, (4) stop at s_2, (5) go to s_3, (6) go back to s_2, and (7) go to s_3. Each WCV in the optimal schedule takes a 5h recharging round, and thus, 5 WCVs are sufficient for recharging all sensors.

This paper provides a clean-state solution to cooperatively schedule WCVs. Our main contributions are fourfold:

- This paper addresses one of the most fundamental problems in WRSNs. A clean problem formulation is provided to minimize the number of WCVs to recharge the sensors.
- For line and cycle spaces, algorithms with approximation ratios of 2 and 4, respectively, are proposed. Optimal solutions are obtained if sensor lifetimes are homogenous.
- In the metric space, an algorithm with an approximation ratio of 2.5 is proposed if sensor lifetimes are homogenous. Through a grouping mechanism, this algorithm is extended to scenarios with heterogenous sensor lifetimes.
- Extensive real data-driven experiments are conducted to evaluate the proposed solutions. The results are shown from different perspectives to provide conclusions.

The remainder of this paper is organized as follows. Section II surveys related works. Section III describes the model, and then, formulates the problem. Section IV studies the problem in line and cycle spaces, respectively. Section V investigates the problem in the metric space. Section VI includes the real data-driven experiments. Section VII concludes the paper.

II. Related Work

WRSNs have been extensively studied in terms of wireless sensor networks, over the past decades. Wang et al. [9, 10] surveyed data collection methodologies, in which data were collected at some sensors and forwarded to base stations for further processing. Research efforts have been made to prolong the sensors' lifetimes. Yao et al. [1] designed an energy-efficient delay-aware lifetime-balancing protocol for WRSNs, by leveraging recent results on open vehicle routing problems. Luo et al. [2] prolonged the sensors' lifetimes by using the shortest path aggregation tree to save energy. However, the sensor battery capacity remains a performance bottleneck for large-scale WRSNs. Peng et al. [11] demonstrated the feasibility of WCVs for WRSNs through building a proof-of-concept prototype. Since wireless power transfer technology has been adopted in WiFi and RFID sensing devices [3, 12], WCVs are acknowledged as solutions to sensor battery problems [13].

The schedules of WCVs are important for recharging the sensors in WRSNs. Xie et al. [4] focused on a WRSN optimization framework by jointly optimizing the traveling path, flow routing, and charging time. However, only one WCV is considered in Xie's work, which may not be feasible for large-scale WRSNs that require cooperative WCV schedules. He et al. [5] proposed an on-demand mobile charging schedule using a simple but efficient nearest-job-next rule with preemptions for the WCV. However, this approach cannot guarantee that each sensor will be recharged before running out of energy. Fu et al. [14] designed an energy synchronized charging protocol for rechargeable WRSNs, using multiple WCVs. Fu's work mainly focused on experimental implementations, and thus, algorithm approximations and properties were not explored. Guo et al. [15] proposed a framework of joint wireless energy replenishment and anchor-point based mobile data gathering in WRSNs, by considering various sources of energy consumption and the time-varying nature of energy replenishment. Their work does not require WCVs to periodically recharge all sensors. Madhajia et al. [16] investigated hierarchical and collaborative WCV schedules in WRSNs. Their algorithms are heuristic, and thus, do not guarantee any bounds. Dai et al. [17] tried to minimize the number of WCVs in WRSNs. However, the approximation ratio of their algorithm is not a constant. This is because their WCVs are not cooperatively scheduled. In contrast, our results are better-bounded.

III. Model, Formulation, and Analysis

A. Model and Formulation

This paper mainly focuses on cooperative WCV schedulings for WRSNs, in line, cycle, and metric spaces (e.g., Euclidean space). Let $S = \{s_1, s_2, ..., s_n\}$ denote the set of sensors, where s_i denotes the i-th sensor. The distance between s_i and s_j is denoted as d_{ij}. Each sensor needs to be recharged before running out of energy, and the lifetime of s_i is denoted as t_i. t_i represents s_i's battery capacity without recharging. To keep s_i running forever, s_i must be periodically recharged within the lifetime, t_i, of its last recharge (or initial time). WCVs are identical, and their maximum speed is denoted as v. WCVs can run at a speed that is less than v and can stop at any place. The range of the wireless power transfer is considered to be ignorable with respect to the distance between sensors. Therefore, WCVs must reach the sensors before recharging them [7]. A sensor can be *fully* and *immediately* recharged once it is reached by a WCV. The recharging time is ignored, since it can be converted to an extra WCV travel distance. For example, if s_i has a recharging time of σ_i, d_{ij} is updated as $d_{ij} + \frac{1}{2}v\sigma_i$ (similar for d_{ji}) to convert the recharging time. The coefficient $\frac{1}{2}$ is used due to round trips. In addition, the recharging of WCVs is not explored in this paper. We assume that there is a special WCV to recharge our WCVs, which can infinitely recharge the sensors.

The objective of this paper is to minimize the number of WCVs, i.e., a minimum WCV problem. The constraint is that all sensors must be periodically recharged before running out of energy. As a result, sensors that are periodically recharged within their lifetimes are running forever. We aim to find a minimum number of WCVs and their recharging routes, such that sensors are running forever. The key challenge comes from the complexity of WCV route schedules, as previously illustrated in Fig. 1. A sensor can be cooperatively recharged by different WCVs in turn, as long as it can run forever.

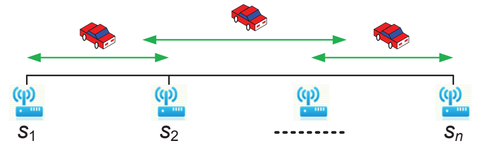


Fig. 2. Example to illustrate the line space.

B. Problem Analysis

This subsection includes the problem hardness analysis:

***Theorem** 1:* The minimum WCV problem is NP-hard.

Proof: The proof is done by the reduction from the travelling salesman problem (TSP), which is a classic NP-hard problem [18]. Given a set of nodes and the distances between nodes, TSP aims at finding out the shortest route that visits each node exactly once and returns to the origin node. Let λ denote the optimal route in the TSP. We reduce each node to a sensor. In the reduction, sensor lifetimes are identical, and are set to be $|\lambda|/v$. $|\lambda|$ is the total length of λ, and v is the speed of the WCVs. If a WCV is assigned to traverse along λ, then all sensors can be periodically recharged before running out of energy. Therefore, the minimum number of WCVs is one. All the other solutions for our problem use more WCVs. As a result, the optimal solution for the TSP reduces to the optimal solution for our problem. Since the TSP reduces to the minimum WCV problem, our problem is NP-hard. ■

The insight of the reduction is that the TSP is a special case of our minimum WCV problem, when one WCV is sufficient for recharging all sensors before their lifetimes expire.

This subsection shows some important observations:

***Lemma** 1:* If a feasible schedule assigns a WCV to run at a speed that is less than v, then it can be converted to a new feasible schedule that assigns all WCVs to run at the speed v and uses no more WCVs than the original schedule.

If a feasible schedule assigns a WCV to run at less than full speed from s_i to s_j, a new feasible schedule can assign this WCV to run full speed to s_j, and then use the extra time to run around s_j. Therefore, Lemma 1 is intuitive. The following paper sets all WCVs to run at the full speed of v. We refer to the optimal schedule as the one that assigns WCVs to run at the speed v, although other optimal schedules may exist.

***Lemma** 2:* In a feasible schedule, each WCV periodically runs along a closed walk among the sensors.

A closed walk consists of a sequence of sensors starting and ending at the same sensor. Note that a closed walk can include a sensor multiple times. Lemma 2 results from the constraint that each sensor is periodically recharged within its lifetime. If a WCV does not run along a closed walk, then it cannot periodically recharge the sensors.

***Lemma** 3:* One feasible schedule, which assigns WCVs to meet each other, can be converted into a new feasible schedule that does not assign WCVs to meet and uses no more WCVs.

If a feasible schedule assigns two WCVs to meet each other, then a new feasible schedule can improve it by a simple swap. These two WCVs can swap their incoming routes just before they meet, leading to a new schedule that uses no more WCVs than the original schedule. Therefore, Lemma 3 validates.

Algorithm 1 Back and Forth (BF)

Input: sensors $\{s_1, s_2, ..., s_n\}$, their locations and lifetimes.
Output: number of WCVs and their route schedules.

```
1: Initialize $i = 1$ (start with the leftmost sensor $s_1$).
2: while $i \leq n$ do
3:     Assign one WCV to go back and forth as far as possible at a full speed to recharge sensors $\{s_i, s_{i+1}, ..., s_{j-1}\}$, such that sensors $\{s_i, s_{i+1}, ..., s_{j-1}\}$ can be periodically recharged by this WCV before their lifetimes:
           $v \times t_k \geq 2\, d_{i,j-1}$ for $\forall k, i \leq k \leq j-1$
           $v \times t_k < 2\, d_{i,j}$ for $\exists k, i \leq k \leq j$
4:     Update the WCV route schedule, and update $i = j$.
5: return number of WCVs and their route schedules.
```

IV. Line Space and Cycle Space

This section investigates the WCV schedules in two special scenarios, line and cycle spaces, to reveal insights for the WCV schedules. In the line space, sensors are deployed along a line, and WCVs can only move along that line. WRSNs deployed in the line space have various monitoring applications for oil pipelines, AC powerlines, and national borderlines [19]. The line space is extended to the cycle space, in which the two ends of the line space are connected. Section V extends line and cycle spaces to the metric space.

A. Line Space

In the line space, sensors are deployed along a line and are denoted by s_1 to s_n from left to right. WCVs are only allowed to move along the line. An example of the line space is shown in Fig. 2. Algorithm 1 is proposed as a clean-state greedy approach to solve the minimum WCV problem in the line space. Algorithm 1 starts with the leftmost sensor, as shown in line 1. Lines 2 to 4 include iterations. In each iteration, one WCV is scheduled to route back and forth between s_i and s_{j-1} to periodically recharge sensors $\{s_i, s_{i+1}, ..., s_{j-1}\}$. The constraint, $v \times t_k \geq 2\, d_{i,j-1}$ for $\forall k, i \leq k \leq j-1$, is satisfied, such that these sensors can run forever. Meanwhile, the greediness means that only one WCV cannot keep $\{s_i, s_{i+1}, ..., s_j\}$ running forever, i.e., $v \times t_k < 2\, d_{i,j}$ for $\exists k, i \leq k \leq j$. Since the current WCV keeps $\{s_i, s_{i+1}, ..., s_{j-1}\}$ running forever, line 4 updates $i = j$ to schedule new WCVs for the remaining sensors on the right. The iteration terminates when all sensors can run forever by recharging. Line 5 returns the number of WCVs and their route schedules. The key idea of Algorithm 1 is to greedily schedule each WCV to recharge as many sensors as possible, such that these sensors can run forever. The time complexity of Algorithm 1 is $O(n)$ by checking the constraint.

We claim that Algorithm 1 is bounded:

***Theorem** 2:* In the line space, Algorithm 1 has an approximation ratio of 2 to the optimal algorithm.

Proof: Let OPT denote the number of WCVs in the optimal algorithm. We can divide the line space into subspaces based on the WCV route composition. Each subspace is defined by a unique set of visited WCVs. An example is shown in Fig. 3, in

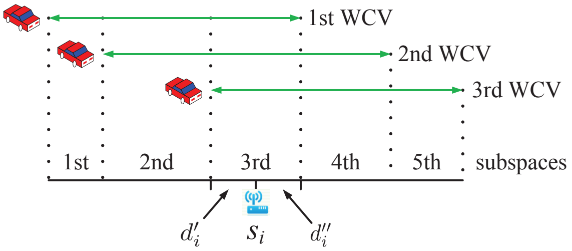


Fig. 3. Example to illustrate the proof of Theorem 2.

which the line space is divided into 5 subspaces by 3 WCVs. Since the route of each WCV has two end points, the number of subspaces is at most 2OPT−1. Subspaces may have different sizes, and more than one sensors may exist in a subspace.

We claim that all sensors in each subspace of the optimal algorithm will not run out of energy if one WCV is assigned to move back and forth at a full speed in the subspace. Let us consider a sensor, s_i, in an arbitrary subspace. Let d'_i and d''_i denote the distances from s_i to the left and right boundaries of its subspace. Let us consider the rightmost (in terms of the geography) WCV that goes through s_i. When the rightmost WCV visits s_i, all the other WCVs in this subspace should be at the left side of the rightmost WCV, since WCVs do not meet each other (Lemma 3). Therefore, when the rightmost WCV leaves s_i and runs to the left, it should be the first WCV that comes back to s_i among all the WCVs in the subspace. s_i is not recharged during the above process, which takes at least $2d'_i/v$. As a result, we have $2d'_i/v \leq t_i$. Similarly, we can get $2d''_i/v \leq t_i$ by considering the leftmost WCV. Hence, s_i can run forever, if one WCV is assigned to run back and forth in its subspace ($2d'_i/v \leq t_i$ and $2d''_i/v \leq t_i$).

Note that Algorithm 1 is optimal under the constraint that all WCVs go back and forth in disjoint routes. Therefore, Algorithm 1 solves the minimum WCV problem with at most 2OPT−1 WCVs, and the proof completes. ■

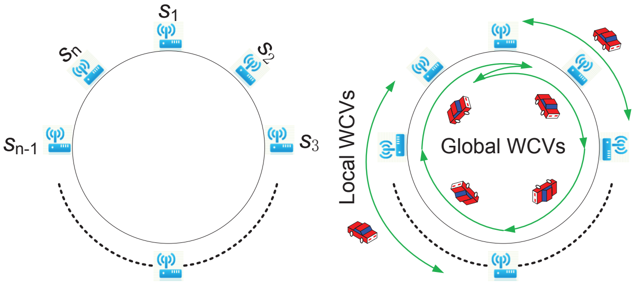
The insight of Theorem 2 is that WCV route overlaps are not efficient in the line space. Therefore, WCVs can be scheduled in disjoint routes. Moreover, Algorithm 1 can be optimal when sensors have identical lifetimes. This is because WCV route overlaps become suboptimal in that scenario.

B. Cycle Space

This subsection extends the line space to the cycle space, in which the two ends of the line space are connected. Sensors are denoted as s_1 to s_n in the clockwise direction (s_1 and s_n are neighbors). WCVs can only move along the cycle. An example is shown in Fig. 4(a). We divide WCVs into the following two types, which are illustrated in Fig. 4(b):

***Definition** 1:* WCVs, which visit the entire cycle space, are global WCVs. All the other WCVs are local WCVs.

A WCV, which visits all sensors, is not necessarily a global WCV. This is because this WCV may not go through the entire cycle space. The route of a global WCV may not be simply traversing the cycle clockwisely or counter-clockwisely. It can be complex, and an example has been illustrated in Fig. 1(d). All five WCVs in Fig. 1(d) are global WCVs.



(a) Cycle space. (b) Two types of WCVs.

Fig. 4. Example to illustrate the cycle space and two types of WCVs.

***Definition** 2:* A local algorithm is defined by assigning only local WCVs to recharge sensors. A global algorithm is defined by assigning only global WCVs to recharge sensors.

***Theorem** 3:* The optimal algorithm is the better one of the optimal local algorithm and the optimal global algorithm.

Theorem 3 appears amazing, but is intuitive if we go back to Lemma 3. This is because a global WCV must meet local WCVs. If an algorithm assigns both local and global WCVs, then it can be improved by Lemma 3. Moreover, we have:

***Theorem** 4:* The optimal global algorithm uses at least half the number of WCVs of the optimal local algorithm.

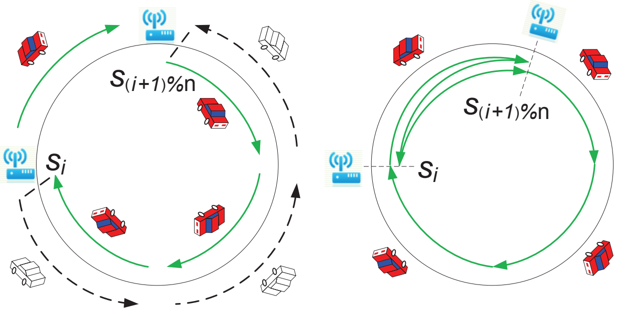
Proof: We prove this by converting the optimal global algorithm into a non-optimal local algorithm. Let us start with an arbitrary i and the space between sensors s_i and $s_{(i+1)\%n}$. The symbol % denotes the modulo operation, i.e., $s_{(n+1)\%n} = s_1$. Based on Definition 1, WCVs in the optimal global algorithm must periodically pass through the space between sensors s_i and $s_{(i+1)\%n}$. Without loss of generality, we assume that a WCV moves clockwisely from s_i to $s_{(i+1)\%n}$, as shown in Fig. 5(a). The converted local algorithm will replace this WCV with an additional set of WCVs to traverse the cycle space counter-clockwisely from s_i to $s_{(i+1)\%n}$. This additional set of WCVs is denoted as white WCVs in Fig. 5(a) and its routes are dotted. WCVs in the converted local algorithm have the same route as those in the optimal global algorithm before reaching s_i and after reaching $s_{(i+1)\%n}$. The above conversion process at most doubles the number of WCVs, since the route length of each WCV is at most doubled.

Note that, in the optimal global algorithm, a WCV may go from s_i to $s_{(i+1)\%n}$, and then, go back from $s_{(i+1)\%n}$ to s_i (this pattern can be even repeated). However, the conversion process remains correct if we assign WCVs (in the additional set) to stop at s_i and $s_{(i+1)\%n}$ to replace the above scenario. By the definition of optimality, the converted local algorithm uses no less WCVs than the optimal local algorithm. Therefore, the optimal global algorithm uses at least half the number of WCVs of the optimal local algorithm. ■

Theorem 4 shows that global WCVs can be approximated or replaced by local WCVs in the cycle space. When combining Theorem 3, it means that a local algorithm can approximate the optimal algorithm. As a result, Algorithm 2 is proposed. It simply tries all the possibilities to break the cycle space to the line space. Algorithm 1 is called to determine the WCV

Algorithm 2 Cycle Break (CB)

```
Input: sensors $\{s_1, s_2, ..., s_n\}$, their locations and lifetimes.
Output: number of WCVs and their route schedules.
for $i = 1$ to $n$ do
    Convert the cycle space to the line space by breaking the space between sensors $s_i$ and $s_{(i+1)\%n}$.
    Call Algorithm 1 to schedule WCVs on the line.
return the best cycle break schedule as the result.
```



(a) Proof of Theorem 4. (b) Proof of Theorem 6.

Fig. 5. Local and global WCV schedules in the cycle space.

schedule after the cycle break. The best cycle break among n possible cycle breaks is returned. Consequently, the time complexity of Algorithm 2 is $o(n^2)$. We have:

***Theorem** 5:* In the cycle space, Algorithm 2 has an approximation ratio of 4 to the optimal algorithm.

Proof: We first claim that Algorithm 2 has an approximation ratio of 2 to the optimal local algorithm. The cycle break may involve one more WCV, when the optimal local algorithm in the cycle space is converted to the optimal algorithm in the line space. Note that Algorithm 1 uses less than double the number of WCVs of the optimal algorithm in the line space (2OPT−1). Hence, our claim is correct (2OPT−1+1=2OPT). According to Theorems 3 and 4, the optimal local algorithm also has an approximation ratio of 2 to the optimal algorithm. Since $2 \times 2 = 4$, Algorithm 2 has an approximation ratio of 4 to the optimal algorithm. ■

The insight of Theorem 5 is that the optimal global algorithm is relatively negligible in the cycle space, while the optimal local algorithm in the cycle space could be approximated by breaking the cycle space to the line space.

C. Optimality under Lifetime Homogeneity

This subsection studies the optimal algorithm in the cycle space, when sensor lifetimes are identical. Let t denote the homogenous sensor lifetime, i.e., $t_i = t$ for all sensors. The idea is to compare the optimal local and global algorithms and choose the better one. As a result, Algorithm 3 is proposed. Line 1 calculates the 1st candidate by calling Algorithm 2 to schedule local WCVs. Line 2 represents the 2nd candidate by scheduling $\lceil \frac{1}{vt} \sum_{i=1}^{n} d_{i,(i+1)\%n} \rceil$ global WCVs to evenly traverse the cycle space. $\lceil \cdot \rceil$ is the round up operator. Line 3 compares the 1st and 2nd candidates, and returns the better one. The time complexity of Algorithm 3 is $O(n^2)$.

Algorithm 3 Compare Two Candidates (CTC)

```
Input: sensors $\{s_1, s_2, ..., s_n\}$, their locations and lifetimes.
Output: number of WCVs and their route schedules.
1: Call Algorithm 2 and record its result as the 1st candidate.
2: Assign $\lceil \frac{1}{vt} \sum_{i=1}^{n} d_{i,(i+1)\%n} \rceil$ WCVs to evenly traverse the cycle space (either clockwisely or counter-clockwisely). Record this WCV schedule as the 2nd candidate.
3: return the better one of the 1st and 2nd candidates.
```

***Theorem** 6:* In the cycle space with identical sensor lifetimes, Algorithm 3 is optimal.

Proof: The proof involves two parts. The first part proves that line 1 in Algorithm 3 is an optimal local algorithm. The second part proves that line 2 in Algorithm 3 is an optimal global algorithm. After proving these two parts, the optimality validates according to Theorem 3.

The first part of the proof is intuitive. In the line space, when sensors have identical lifetimes, it is suboptimal to schedule WCVs in overlapped intervals (otherwise a swap can lead to a better schedule). When sensors have identical lifetimes, Algorithm 1 is optimal in the line space, and thus, Algorithm 2 is an optimal local algorithm in the cycle space.

The second part of the proof is shown in Fig. 5(b). Based on Lemma 2, the route of each global WCV is a closed walk that visits the entire cycle space. Let m_i denote the number of times that s_i appears in the route of each global WCV. In Fig. 5(b), we have $m_i = 2$ and $m_{(i+1)\%n} = 2$. Each global WCV only changes its direction at a sensor, and does not stop at a sensor, due to the lifetime homogeneity. Let $m = \min_i m_i$. Since each sensor appears in the route of a WCV at least m times, the route length is at least $m \times \sum_{i=1}^{n} d_{i,(i+1)\%n}$ (m times the cycle circumference). During one closed walk, a sensor is charged for at least $\frac{m}{vt} \sum_{i=1}^{n} d_{i,(i+1)\%n}$ times. Let us consider the sensor s_j that appears the least in the closed walk. During one closed walk of a WCV, s_j is charged at most m times. Therefore, at least $\lceil \frac{1}{m} \frac{m}{vt} \sum_{i=1}^{n} d_{i,(i+1)\%n} \rceil$ WCVs are needed (lower bound of the optimal global algorithm). Line 2 in Algorithm 3 is a feasible schedule that can keep sensors running forever. Since it uses the same number of WCVs as the lower bound, it is an optimal global algorithm. ■

The key insight of Theorem 6 is that, when sensor lifetimes are homogenous, the optimal local algorithm schedules WCVs in disjoint routes, and the optimal global algorithm schedules WCVs to evenly traverse the cycle space. Based on Theorem 3, the optimal algorithm is the better of the optimal local algorithm and the optimal global algorithm.

V. Metric Space

The previous section investigates the WCV schedules in line and cycle spaces, which may not be feasible for general WRSNs. Consequently, this section extends previous results to the metric space [20]. Note that the metric space includes the three-dimensional Euclidean space, and thus, our study can be applied to practical WRSNs with WCVs.

Algorithm 4 Cycles in Minimum Forest (CMF)

```
Input: sensors $\{s_1, s_2, ..., s_n\}$, their locations and lifetimes.
Output: number of WCVs and their route schedules.
1: for $k = 0$ to $n - 1$ do
2:   Find minimum distance forest for $\{s_1, s_2, ..., s_n\}$ with
     $k$ undirected edges. $n - k$ connected sensor components
     are generated, and are denoted as $C_1$, $C_2$, ..., $C_{n-k}$.
3:   for each connected sensor component $C_i$ do
4:     Use the Christofides algorithm to resolve the TSP in
       $C_i$. Convert the TSP route to a cycle space.
5:     Call Algorithm 3 to recharge sensors in $C_i$.
6:   Record the current WCV schedule as a candidate.
7: return the best one among $n$ candidate WCV schedules.
```

A. WCV Scheduling under Lifetime Homogeneity

This subsection studies the WCV schedules in the metric space, under the sensor lifetime homogeneity (i.e., $t_i = t$). The key idea is to control the total lengths of WCV routes by using minimum distance forests among sensors. As a result, Algorithm 3 is extended to Algorithm 4 for the metric space. In lines 1 to 6, Algorithm 4 includes an iteration to go through all possible minimum distance forests. In each iteration, it builds a minimum distance forest for $\{s_1, s_2, ..., s_n\}$ with k undirected edges. $n - k$ connected sensor components are generated, and are denoted as C_1, C_2, ..., C_{n-k}. The minimum distance forest can be obtained by the Kruskal's algorithm with an early termination (i.e., terminate the greedy minimum spanning tree algorithm when it has chosen the first k edges). In lines 3 to 5, each connected sensor component is processed as a cycle space. The Christofides algorithm is used to find an approximated route for the travelling salesman problem (TSP) in C_i, with an approximation ratio of 1.5 in terms of the route length. It converts a connected sensor component into a cycle space, which is in turn solved by Algorithm 3. The WCV schedule is recorded in line 6. Finally, line 7 returns the best schedule among all possible minimum distance forests.

An example is shown in Fig. 6 to illustrate Algorithm 4. The scenario sets $n = 7$ with $k = 5$. Line 2 finds the minimum distance forest for the sensors by adding $k = 5$ edges (s_1-s_2, s_1-s_3, s_4-s_5, s_5-s_6, and s_6-s_7). This forest is denoted by black lines, and can be obtained by terminating the greedy minimum spanning tree algorithm when it has chosen the first $k = 5$ edges. The forest includes $n - k = 2$ connected sensor components (C_1 and C_2). For each component, the Christofides algorithm is used to compute a TSP route, which is denoted by the dotted line. The TSP route in each component is regarded as a cycle space (repeated sensors are removed). Algorithm 3 is called to schedule WCVs in the cycle space. The best result among different k (different forests) is returned. Since the Christofides algorithm takes $O(n^3)$, Algorithm 4 takes $O(n^4)$ by traversing k from 0 to $n - 1$. We have:

Theorem 7: In the metric space with homogeneous sensor lifetimes, Algorithm 4 has an approximation ratio of 2.5 to the optimal algorithm.

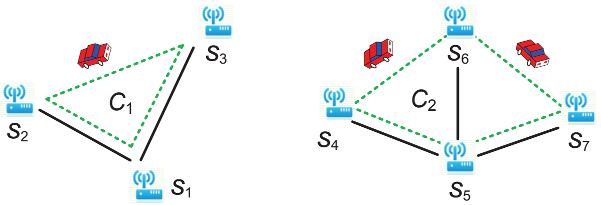


Fig. 6. An example to illustrate Algorithm 4.

Proof: Let CMF and OPT be the numbers of WCVs used by Algorithm 4 and the optimal algorithm, respectively. Let us consider the connected components in the optimal algorithm. We say that two sensors share an undirected edge if a WCV goes from one sensor to the other within t. Suppose that the optimal algorithm has $n - k^*$ components, which are denoted as C_1^*, C_2^*, ..., $C_{n-k^*}^*$. Let $|C|$ denote the total length of edges in C. We claim that the number of WCVs used by the optimal algorithm is at least $\sum_{i=1}^{n-k^*} \lceil \frac{1}{vt}|C_i^*| \rceil$. This is because the WCV route is a closed walk, and WCVs periodically go among sensors to recharge them (Lemma 2).

Since Algorithm 4 checks all possible minimum distance forests, it will go through the case $k = k^*$. Since the forest is minimal, we have $\sum_{i=1}^{n-k^*} |C_i| \leq \sum_{i=1}^{n-k^*} |C_i^*|$. Let C_i' denote the cycle space corresponding to C_i, and let $|C_i'|$ denote its cycle circumference. Since the Christofides algorithm has an approximation ratio of 1.5 with respect to the route length, we have $\sum_{i=1}^{n-k^*} |C_i'| \leq 1.5 \sum_{i=1}^{n-k^*} |C_i|$. Since sensors have identical lifetimes, Algorithm 3 can optimally schedule WCVs for each cycle space. For C_i', at most $\lceil \frac{1}{vt}|C_i'| \rceil$ WCVs are used by scheduling only global WCVs in the cycle space. If we combine the above results, we have:

$$\begin{aligned} \text{CMF} &\leq \sum_{i=1}^{n-k^*} \lceil \frac{1}{vt}|C_i'| \rceil \leq \sum_{i=1}^{n-k^*} \left(\frac{1}{vt}|C_i'| + 1 \right) \\ &\leq \sum_{i=1}^{n-k^*} \left(\frac{1.5}{vt}|C_i| + 1 \right) \leq \sum_{i=1}^{n-k^*} \left(\frac{1.5}{vt}|C_i^*| + 1 \right) \qquad (1) \end{aligned}$$

Since the optimal algorithm assigns at least one WCV for each component, we have $\lceil \frac{1}{vt}|C_i^*| \rceil \geq 1$:

$$\begin{aligned} \text{CMF} &\leq \sum_{i=1}^{n-k^*} \left(\frac{1.5}{vt}|C_i^*| + 1 \right) \leq \sum_{i=1}^{n-k^*} \left(1.5 \lceil \frac{1}{vt}|C_i^*| \rceil + 1 \right) \\ &\leq \sum_{i=1}^{n-k^*} \left(2.5 \lceil \frac{1}{vt}|C_i^*| \rceil \right) \leq 2.5 \sum_{i=1}^{n-k^*} \lceil \frac{1}{vt}|C_i^*| \rceil \qquad (2) \end{aligned}$$

Since $\text{OPT} \geq \sum_{i=1}^{n-k^*} \lceil \frac{1}{vt}|C_i^*| \rceil$, Eq. 2 can be simplified as $\text{CMF} \leq 2.5 \times \text{OPT}$. Therefore, the proof completes. ■

The insight of Theorem 7 is that the number of WCVs in the optimal algorithm can be lower-bounded through the total lengths of its WCV routes. Therefore, Algorithm 4 guarantees an approximation ratio by controlling the total lengths of WCV routes using the minimum distance forest.

Algorithm 5 Group Sensors by Lifetimes (GSL)

Input: sensors $\{s_1, s_2..., s_n\}$, their locations and lifetimes.
Output: number of WCVs and their route schedules.

```
1: for $i = 1$ to $\lfloor \log_2 \frac{\max_i t_i}{\min_i t_i} \rfloor + 1$ do
2:     Find the $i$-th group of sensors that satisfy:
           $\{\, s_j \,|\, 2^{i-1} \times \min_i t_i \le t_j < 2^i \times \min_i t_i$ and $s_j \in S\}$.
3:     For sensors in the $i$-th group, approximate all of their
       lifetimes to be $2^{i-1} \times \min_i t_i$ (convert to homogeneity).
4:     Call Algorithm 4 to recharge sensors in the $i$-th group.
5: return the WCV schedules in all sensor groups.
```

B. WCV Scheduling under Lifetime Heterogeneity

This subsection explores the WCV schedules in the metric space under the sensor lifetime heterogeneity. Consequently, Algorithm 5 is proposed to extend Algorithm 4. The key idea is to group sensors according to their lifetimes, i.e., sensors with similar lifetimes are grouped together. In each sensor group, sensor lifetimes are approximated to be the same, and then, Algorithm 4 is called to recharge sensors in this group. The subtle design is the group criterion in line 1 of Algorithm 5. $\lfloor \cdot \rfloor$ is the round down operator. Let $\min_i t_i$ and $\max_i t_i$ be the minimum and maximum sensor lifetimes in S, respectively. The i-th sensor group is consists of sensors with lifetimes from $2^{i-1} \times \min_i t_i$ to $2^i \times \min_i t_i$, i.e., sensors are grouped exponentially with respect to their lifetimes. Note that each sensor belongs to exactly one sensor group. In line 2 of Algorithm 5, lifetimes of sensors in the i-th group are approximated to their lower bounds, $2^{i-1} \times \min_i t_i$. Such approximations guarantee that sensors can be recharged within their real lifetimes. Sensors in the i-th group are approximated to have identical lifetimes. Consequently, Algorithm 4 is called to recharge them in line 4. Finally, the WCV schedules in all sensor groups are returned together in line 5.

The time complexity of Algorithm 5 is $O(n^4)$, which is the same as Algorithm 4. This is because the group mechanism in line 2 can reduce the complexity of Algorithm 4 in line 4. Let n_i denote the number of sensors in the i-th group, and we have $\sum_i n_i^4 \le (\sum_i n_i)^4 = n^4$. The group mechanism in line 2 takes only $O(n \log n)$ by sorting sensors lifetimes. If a group does not include a sensor, it is ignored. We have:

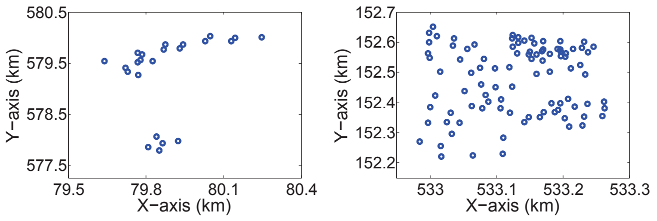
***Theorem** 8:* In the metric space with heterogenous sensor lifetimes, Algorithm 5 guarantees an approximation ratio of $5\lfloor \log_2 \frac{\max_i t_i}{\min_i t_i} \rfloor + 5$ to the optimal algorithm.

Proof: Let GSL and OPT denote the number of WCVs used by Algorithm 5 and the optimal algorithm, respectively. Let GSL_i denote the number of WCVs in the i-th sensor group of GSL. By definition, we have $\text{GSL} = \sum_i \text{GSL}_i$. Let OPT_i denote the number of WCVs used by the optimal algorithm for only sensors in the i-th group. Since OPT_i does not recharge all sensors, $\text{OPT}_i \le \text{OPT}$. We claim that $\text{GSL}_i \le 2.5 \times 2 \times \text{OPT}_i$. This is because Algorithm 4 has an approximation ratio of 2.5, and the sensor lifetime approximation in line 3 at most doubles the number of WCVs (the lifetime of each sensor is at most halved by line 3). Since Algorithm 5 has at most $\lfloor \log_2 \frac{\max_i t_i}{\min_i t_i} \rfloor + 1$ groups, we have:

$$\begin{aligned}\text{GSL} &= \sum_i \text{GSL}_i \le 5 \sum_i \text{OPT}_i \le 5 \sum_i \text{OPT} \\ &\le 5 \times (\Big\lfloor \log_2 \frac{\max_i t_i}{\min_i t_i} \Big\rfloor + 1) \times \text{OPT} \qquad (3)\end{aligned}$$

The proof completes. ■

The key insight of Theorem 8 is that sensors are divided into a limited number of groups. Sensors in the same group have similar lifetimes, and thus, can be resolved by Algorithm 4. Theorem 8 can be further improved by incorporating the sensor lifetime distribution. For example, if t_i is exponentially distributed, then $\lfloor \log_2 \frac{\max_i t_i}{\min_i t_i} \rfloor + 1$ becomes a constant.



(a) GSBD dataset.

(b) LCUE dataset.

Fig. 7. Top view of sensor locations in the GSBD and LUCE datasets.

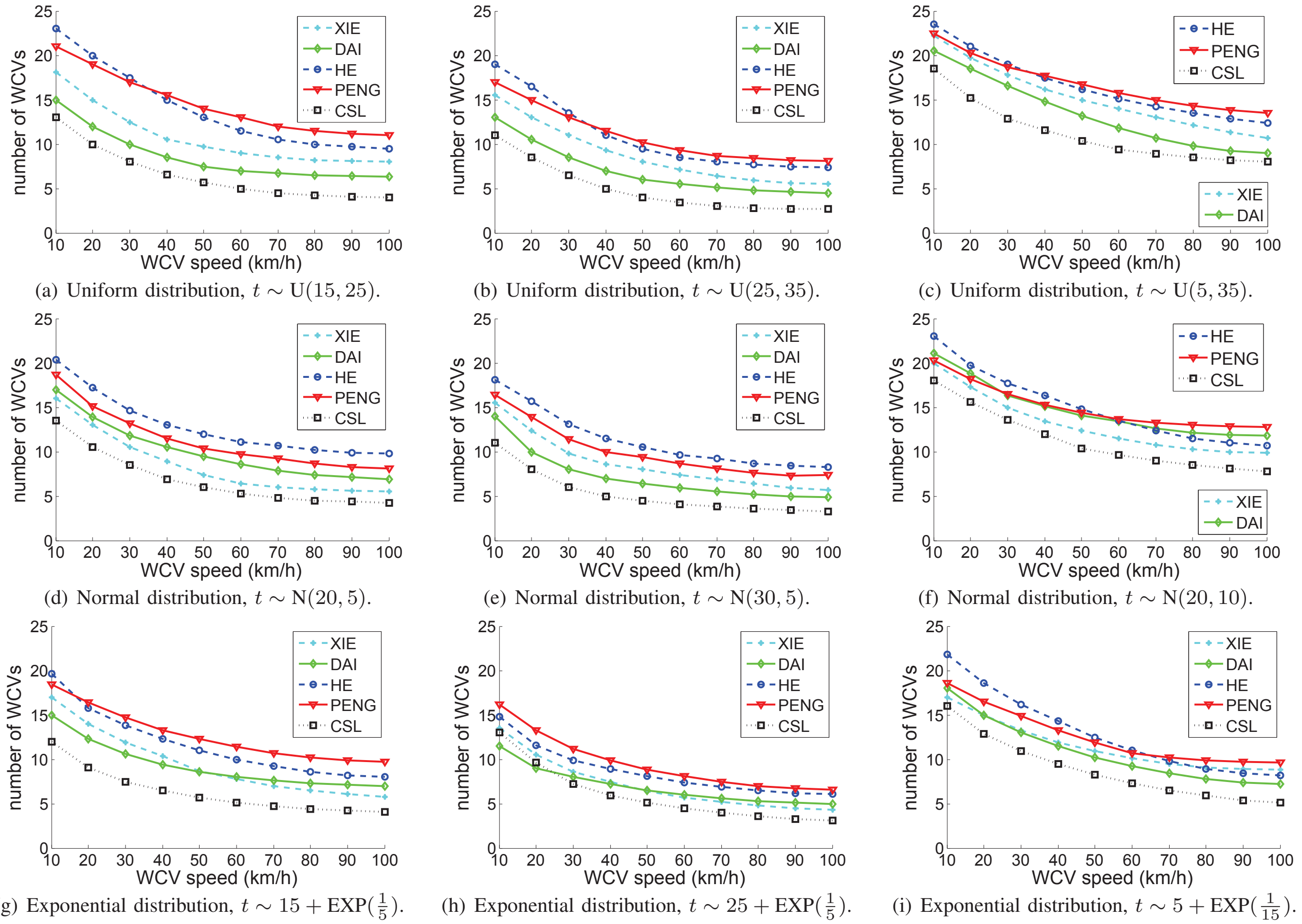
VI. Real Data-Driven Experiments

A. Settings and Comparison Algorithms

Our experiments are based on two environmental sensing datasets that are listed in [21]. The first dataset is the Grand-St-Bernard Deployment (GSBD). It is a SensorScope network deployed between Switzerland and Italy. GSBD includes 23 sensors, and their locations are shown in Fig. 7(a). The second dataset is the Lausanne Urban Canopy Experiment (LUCE). It is a measurement campaign on the EPFL campus that is used for micrometeorology and atmospheric transport. LUCE includes 94 sensors, and their locations are shown in Fig. 7(b). We mainly focus on the number of WCVs with respect to the WCV speed, under three different distributions of sensor lifetimes. The WCV speed is tuned from 10 km/h to 100 km/h. Results are averaged over 1,000 times for smoothness.

Algorithms 1 to 4 are not compared due to their prerequisites. Algorithm 5 is denoted by CSL, and is compared with:

- XIE [4] uses linear programming to near-optimally schedule only one WCV to recharge sensors. This approach is iteratively applied until all sensors can run forever.
- DAI [17] schedules WCVs according to vehicle routing problems. It can construct a forest among sensors, and assigns WCVs to each tree in the forest. Sensors are not cooperatively recharged by multiple WCVs.
- HE [5] is a on-demand approach, in which each WCV prioritizes the nearest sensor that is close to its lifetime. This approach is also iteratively applied until all sensors can be periodically recharged within their lifetimes.
- PENG [11] is a greedy approach that schedules WCVs according to sensor lifetimes. Sensors with shorter lifetimes are prioritized by WCVs.



(a) Uniform distribution, $t \sim \mathrm{U}(15, 25)$.

(b) Uniform distribution, $t \sim \mathrm{U}(25, 35)$.

(c) Uniform distribution, $t \sim \mathrm{U}(5, 35)$.

(d) Normal distribution, $t \sim \mathrm{N}(20, 5)$.

(e) Normal distribution, $t \sim \mathrm{N}(30, 5)$.

(f) Normal distribution, $t \sim \mathrm{N}(20, 10)$.

(g) Exponential distribution, $t \sim 15 + \mathrm{EXP}(\frac{1}{5})$.

(h) Exponential distribution, $t \sim 25 + \mathrm{EXP}(\frac{1}{5})$.

(i) Exponential distribution, $t \sim 5 + \mathrm{EXP}(\frac{1}{15})$.

Fig. 8. Experimental results in the GSBD dataset.

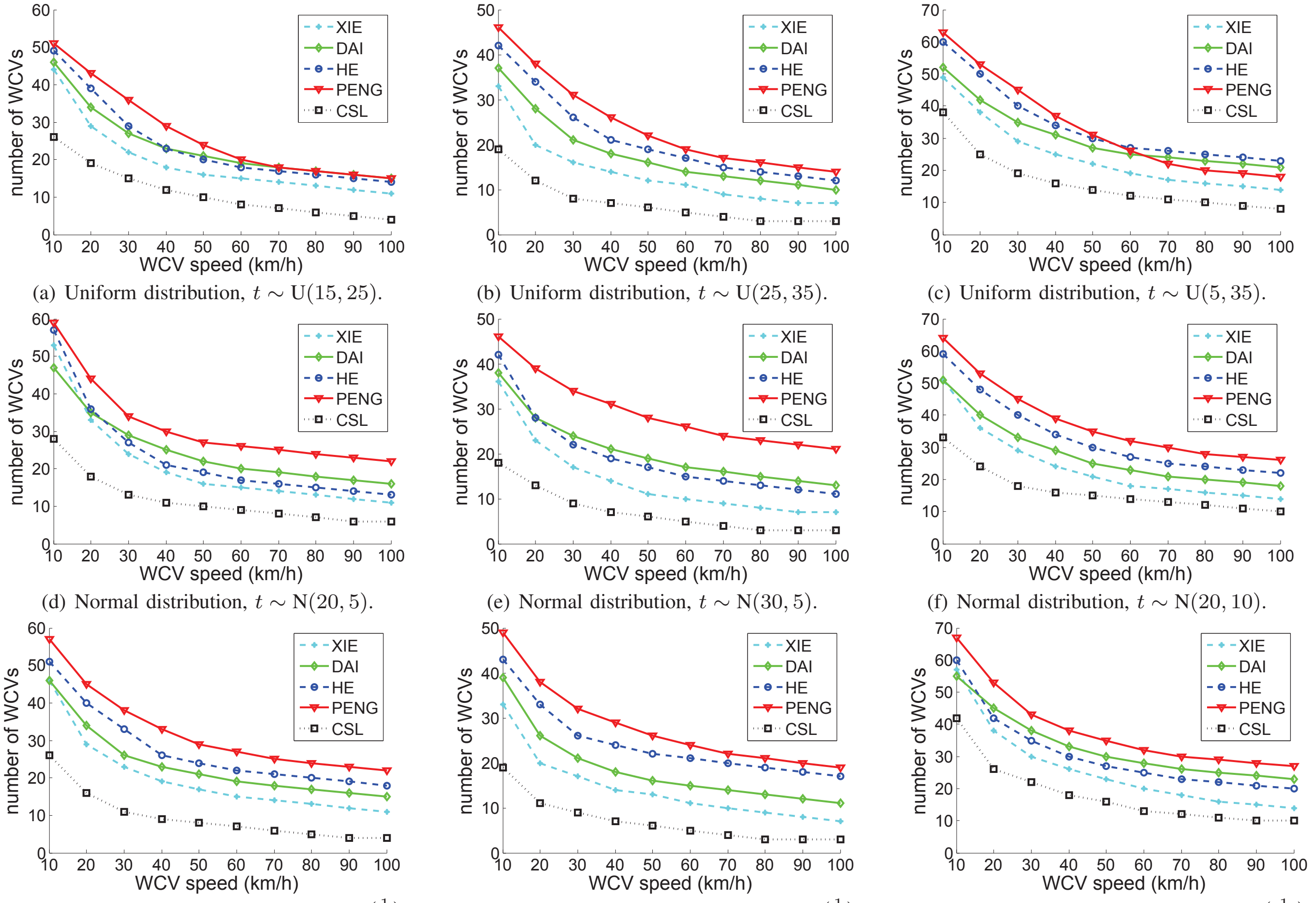
B. Evaluation Results

Our experiments focus on the number of WCVs with respect to the WCV speed, under three sensor lifetime distributions [22, 23]: uniform distribution (with lower and upper bounds), normal distribution (with mean and variance), and exponential distribution (with its exponent parameter). A smaller number of WCVs represents a better result.

The experimental results for the GSBD dataset are shown in Fig. 8. For three rows of subfigures, sensor lifetimes follow three different distributions, respectively (uniform, normal, and exponential distributions in sequence). The sensor lifetime distributions for the first and second columns of subfigures have the same variance, but different mean values. In contrast, the sensor lifetime distributions for the first and third columns of subfigures have the same mean value, but different variances. It can be seen that CSL outperforms the other algorithms, since it schedules WCVs cooperatively. HE and PENG have the worst performances due to their greediness (trapped in local optima). XIE and DAI do not schedule WCVs cooperatively, and thus, cannot efficiently minimize the number of WCVs. A greater WCV speed leads to a smaller number of WCVs for all algorithms, since each WCV can recharge more sensors with a greater speed. By comparing the first and second columns of subfigures in Fig. 8, we can conclude that a shorter average sensor lifetime leads to a larger number of necessary WCVs. This is because sensors need to be recharged more frequently if their lifetimes are shorter. By comparing the first and third columns of subfigures in Fig. 8, we can further conclude that a greater variation in sensor lifetimes also leads to a larger number of needed WCVs. This means that sensors with shorter lifetimes have more impacts on WCV schedules than those with longer lifetimes. The experimental results for the LUCE dataset are shown in Fig. 9, which is similar to the GSBD dataset. However, the performance improvement of CSL is more significant in LUCE than GSBD, since LUCE includes more sensors. By comparing Figs. 8 and 9, CSL has the best asymptotical performance when WRSNs scale up.

VII. Conclusion

This paper studies cooperative WCV schedules in WRSNs, in order to save the cost of sensor recharging. The objective is to minimize the number of WCVs, under the constraint that sensors can run forever by being periodically recharged. Our problem is NP-hard. Based on different insights, approximation algorithms are proposed for line, cycle, and metric spaces, respectively. Real data-driven experiments demonstrate the efficiency and effectiveness of the proposed algorithms.



(a) Uniform distribution, $t \sim \mathrm{U}(15, 25)$.

(b) Uniform distribution, $t \sim \mathrm{U}(25, 35)$.

(c) Uniform distribution, $t \sim \mathrm{U}(5, 35)$.

(d) Normal distribution, $t \sim \mathrm{N}(20, 5)$.

(e) Normal distribution, $t \sim \mathrm{N}(30, 5)$.

(f) Normal distribution, $t \sim \mathrm{N}(20, 10)$.

(g) Exponential distribution, $t \sim 15 + \mathrm{EXP}(\frac{1}{5})$.

(h) Exponential distribution, $t \sim 25 + \mathrm{EXP}(\frac{1}{5})$.

(i) Exponential distribution, $t \sim 5 + \mathrm{EXP}(\frac{1}{15})$.

Fig. 9. Experimental results in the LUCE dataset.

VIII. Acknowledgments

This research was supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, and CNS 1439672.

References

[1] Y. Yao, Q. Cao, and A. V. Vasilakos, "Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks," *IEEE/ACM ToN*, 2015.

[2] D. Luo, X. Zhu, X. Wu, and G. Chen, "Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks," in *IEEE INFOCOM*, 2011.

[3] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-fi backscatter: internet connectivity for RF-powered devices," *ACM SIGCOMM Computer Communication Review*, 2015.

[4] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "Multi-node wireless energy charging in sensor networks," *IEEE/ACM ToN*, 2015.

[5] L. He, L. Kong, Y. Gu, J. Pan, and T. Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE TMC*, 2015.

[6] N. Wang and J. Wu, "Trajectory scheduling for timely data report in underwater wireless sensor networks," in *IEEE GLOBECOM*, 2015.

[7] S. Ho, J. Wang, W. Fu, and M. Sun, "A comparative study between novel witricity and traditional inductive magnetic coupling in wireless charging," *IEEE TMAG*, 2011.

[8] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? A measurement study on GreenOrbs," *IEEE TPDS*, 2013.

[9] F. Wang and J. Liu, "Networked wireless sensor data collection: issues, challenges, and approaches," *IEEE Communications Surveys & Tutorials*, 2011.

[10] N. Wang and J. Wu, "A general data and acknowledgement dissemination scheme in mobile social networks," in *IEEE MASS*, 2014.

[11] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "Prolonging sensor network lifetime through wireless charging," in *IEEE RTSS*, 2010.

[12] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *USENIX NSDI*, 2014.

[13] F. Akhtar and M. H. Rehmani, "Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review," *Renewable and Sustainable Energy Reviews*, 2015.

[14] L. Fu, H. Liu, L. He, Y. Gu, P. Cheng, and J. Chen, "Demo: an energy synchronized charging protocol for rechargeable wireless sensor networks," in *ACM MobiHoc*, 2014.

[15] S. Guo, C. Wang, and Y. Yang, "Joint mobile data gathering and energy provisioning in wireless rechargeable sensor networks," *IEEE TMC*, 2014.

[16] A. Madhja, S. Nikoletseas, and T. P. Raptis, "Hierarchical, collaborative wireless charging in sensor networks," in *IEEE WCNC*, 2015.

[17] H. Dai, X. Wu, L. Xu, G. Chen, and S. Lin, "Using minimum mobile chargers to keep large-scale wireless rechargeable sensor networks running forever," in *IEEE ICCCN*, 2013.

[18] S. O. Gharan, A. Saberi, and M. Singh, "A randomized rounding approach to the traveling salesman problem," in *IEEE FOCS*, 2011.

[19] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks," *Journal of Intelligent & Robotic Systems*, 2014.

[20] Y. Bartal, L.-A. Gottlieb, and R. Krauthgamer, "The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme," in *ACM STOC*, 2012.

[21] [Online]. Available: http://lcav.epfl.ch/cms/lang/en/pid/86035

[22] W. Chang and J. Wu, "Progressive or conservative: Rationally allocate cooperative work in mobile social networks," *IEEE TPDS*, 2015.

[23] L. Wu, X. Du, and X. Fu, "Security threats to mobile multimedia applications: Camera-based attacks on mobile phones," *IEEE Communications Magazine*, 2014.