Mobile-to-Mobile Energy Replenishment in Mission-Critical Robotic Sensor Networks

Liang He[1], Peng Cheng[2,1], Yu Gu[1], Jianping Pan[3], Ting Zhu[4], Cong Liu[5]
[1]Singapore University of Technology and Design, Singapore [2]Zhejiang University, Zhejiang, China
[3]University of Victoria, BC, Canada [4]Binghamton University, NY, US [5]University of Texas, Dallas, US

***Abstract*—Recently, much research effort has been devoted to employing mobile chargers for energy replenishment of the robots in robotic sensor networks. Observing the discrepancy between the charging latency of robots and charger travel distance, we propose a novel tree-based charging schedule for the charger, which minimizes its travel distance without causing the robot energy depletion. We analytically evaluate its performance and show its closeness to the optimal solutions. Furthermore, through a queue-based approach, we provide theoretical guidance on the setting of the remaining energy threshold at which the robots request energy replenishment. This guided setting guarantees the feasibility of the tree-based schedule to return a depletion-free charging schedule. The performance of the tree-based charging schedule is evaluated through extensive simulations. The results show that the charger travel distance can be reduced by around 20%, when compared with the schedule that only considers the robot charging latency.**

I. Introduction

Robotic sensor networks have been increasingly adopted in applications such as environmental monitoring for detecting harmful algal blooms [1], concentration of chemical substances [2], aquatic diffusion process [3], [4], and search and rescue systems [5]. While efficient in performance, the limited energy supply of robots impedes the sustainable operation of robotic sensor networks [6]. Various solutions have been proposed to solve this energy issue in recent years. Silverman et al. propose to deploy fixed charging stations to replenish the robot energy [7]. However, such a mechanism requires additional robot movement, which may conflict with their regular operations. Energy harvesting has also been proposed for the robot energy replenishment, which requires no additional movements [8], [9]. However, the harvested energy is highly unpredictable [10], [11], which significantly affects system reliability.

Adopting mobile chargers to carry out the energy replenishment of robots is a promising approach to provide reliable and sustainable energy supply for robots, and has attracted much research attention [12]–[14]. Unfortunately, most existing schemes require the robot to coordinate with the mobile charger for energy replenishment. Such coordination may cause the robot to be interfered from their normal operations, which is undesirable and even infeasible under scenarios where robots are designated to perform critical tasks such as life search and rescue [5] and diffusion process tracking [3]. Due to this limitation, previous schemes with mobile chargers in robotic sensor networks may not be employed for the mission-critical applications.

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

Motivated by the above observations, in this paper, we focus on the *mission-critical mobile-to-mobile* charging scenario, in which the energy replenishment of the robots cannot interfere with their normal operations, e.g., due to the emergency of the robot's rescue tasks. Consequently, it is solely the responsibility of the charger to rendezvous and travel along with the to-be-charged robot to accomplish the charging task. Specifically, by observing the discrepant requirements between robots (short charging latency) and chargers (short travel distance), we propose a novel charging schedule design for the charger that jointly optimizes the energy replenishing process according to these two requirements. By eliminating the interference on robots due to the charging requirements, we expect our solution to be a better alternative for mission-critical robotic sensor networks.

Our major contributions in this paper include:

- We introduce the concept of mission-critical mobile-to-mobile energy replenishment in robotic sensor networks. To the best of our knowledge, this is the first design that jointly optimizes the energy replenishment for both the robots and chargers.
- We transform the mission-critical mobile-to-mobile charging problem to a shortest-path problem, and design an efficient charging schedule that minimizes the charger travel distance without causing the robot energy depletion. The performance of the proposed schedule is analytically evaluated with regard to the optimal solutions.
- To assist system implementation, we present a queue-based analytical approach that provides guidance on setting the robot's remaining energy threshold (according to which the robots send out requests for energy replenishment). We show that this approach can guarantee the feasibility for the proposed schedule to return a depletion-free solution.

The paper is organized as follows. We present the problem statement in Section II. The proposed charging schedule is presented in Section III, and the guidance on setting the remaining energy threshold is presented in Section IV. Section V evaluates the performance of the charging schedule and the guidance. Section VI reviews the literature, and we conclude this paper in Section VII.

II. Preliminaries

A. Mobile-to-Mobile Energy Replenishment

The advancement of energy-transferring technologies has made the mobile-to-mobile charging effective and desirable.

For example, Zhu et al. has prototyped an energy-sharing system with capacitor-array powered MicaZ motes [8], and the empirical measurements in [10] show that the time to charge a $300\ F$ super-capacitor to a voltage of $2.5\ V$ is in the order of tens of seconds. At the larger scale, the charging power of the current charging stations for electric vehicles can easily reach up to $100\ kW$, e.g., the CT3000 charging station [15]. These high power charging technologies significantly facilitate the adoption of the mobile-to-mobile energy replenishment in the near future.

B. System Model

We consider the scenario where n robots are deployed to carry out tasks such as field exploring and monitoring with travel speed v_r. A mobile charger with controlled mobility and speed v_c ($v_c > v_r$) is employed to replenish the robot energy supply via direct-contact charging technologies [16] as in iRobot Roomba [17].

Robots actively monitor their remaining energy levels [18], [19]. Specifically, let C denote the robot energy capacity and θ denote the remaining energy threshold that triggers the energy replenishment requests. A robot will send out its charging request when its remaining energy level is below θC. The delivery of the charging requests to the charger can be accomplished through techniques such as satellite and wireless ad hoc communications [20], [21]. The time for the delivery of the charging requests is assumed to be negligible when compared with the travel time of the robots [22].

The charger maintains a buffer to store the received charging requests, and travels to rendezvous with the corresponding robots to replenish their energy supply. The charger power capacity is assumed to be enough for the energy replenishment process in consideration [23], [24]. After the rendezvous between the charger and the robot, the time required to accomplish the charging of the robot can be determined based on the robot charging profile and its remaining energy level. We utilize the time to fully charge an energy depleted robot T_c as the required charging time to simplify the presentation.

We focus on the *mission-critical mobile-to-mobile energy replenishment*, which imposes two unique requirements on the energy replenishment process. First, the operation schedule of the robots should not be modified to facilitate their energy replenishment, and thus it is solely the responsibility of the charger to accomplish the charging tasks; second, the charger should reduce its speed to v_r and travel along with the to-be-charged robot during the charging process after their rendezvous.

Nowadays, many *off-the-shelf* robots can achieve hours of continuous operation time, e.g., our customized robot powered by two $1,000\ mAh$ Lithium-Polymer battery connected in series, can continuously operate for up to four hours with a single charge. Similar operation times have been reported by the Pioneer 3-AT robot (two to four hours) [25] and the Scout robot (around two hours) [26]. Therefore in a typical robotic sensor network, a robot normally would not send out its charging request so frequently. As a result, we focus on the light charging request intensity scenario in this work.

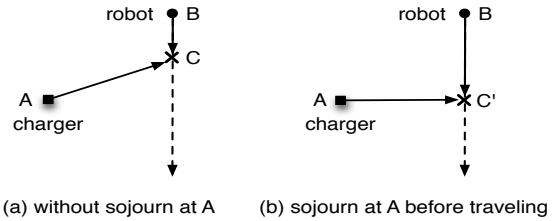

Fig. 1. Postponing charging service achieves a shorter charger travel distance.

C. Problem Statement

For the sustainable system operation, the *charging latency* of the robots, defined as *the time from which a robot sends out its charging request to the time its energy supply is replenished*, has to be controlled within a certain range to avoid their energy depletion. On the other hand, the *travel distance* of the charger to carry out the energy replenishment can be viewed as the *overhead* to provide the charging service, and should be minimized for the interests of the charger.

In the context where the targets are stationary devices, the device charging latency and the charger travel distance can be inferred from each other [27]. However, these two metrics conflict with each other in the mobile-to-mobile charging context. More importantly, we observe that in order to minimize the charging overhead, *it is not necessary for the mobile charger to always provide the charging service continuously, and sometimes intelligently postponing its charging service can improve the charging process in terms of its travel distance*. An illustration of this observation is shown in Fig. 1, where the charger and the to-be-charged robot is located at A and B, respectively. We can see that by delaying the rendezvous time of the charger and the robot (and thus increasing the charging latency), the charger travel distance can be reduced from $|\overline{AC}|$ in Fig. 1(a) to $|\overline{AC'}|$ in Fig. 1(b).

This observation prompts us to determine *when it is beneficial for the charger to postpone its charging service, and for how long this postponement should last*. To answer these questions, we design a novel charging schedule for the charger that intelligently postpones its charging service without causing robot energy depletion. Furthermore, we also provide guidance on how to set the remaining energy level θ for the robots to request energy through a queue-based analytical approach.

III. Tree-based Charging Schedule

In this section we introduce a tree-based schedule scheme for the charger, which minimizes the charger travel distance without causing the robot energy depletion. Our approach is to construct a weighted travel tree based on the planned trajectory of the robots, and then transform the charging schedule problem to a *Single-Source Shortest Path Problem* [28], based upon which a feasible and efficient charging schedule can be obtained.

A. Travel Profiles of the Robots

Normally, the travel trajectories of in-operation robots have been planned for certain time into the future, e.g., in diffusion process tracking [3]. The mobile charger needs these planned trajectories (referred to as the robot *travel profiles*) to carry out the energy replenishment of robots, which can be made available through the communications between them. Consider the scenario where m requests exist in the charger's buffer when it needs to make its charging schedule at location a_0, with corresponding requesting robots $\{r_1, r_2, \cdots, r_m\}$ according to their arrival sequence. Discretizing the time into short slots of duration δ, the travel profile of robot r_i can be represented as

$$\mathcal{P}_i = \{(a_i^j, j\delta)\}\ (j \in [0, \frac{T_i}{\delta}]),$$

where T_i is the time period during which the trajectory of r_i has been planned (the current time is treated as 0 for notation convenience), and a_i^j is its location at time $j\delta$ [1]. The discretization granularity δ is a critical factor that trades off between the schedule performance and complexity, which we will further investigate in Section V-B. Our results show that the effect of reducing δ on shortening the travel distance is limited.

B. Estimate the Rendezvous Deadline

For any requesting robot, the charger can estimate its remaining operation time before energy depletion. Denote μ as the busy state energy consumption rate of the robots. Their worst-case (shortest) remaining operation time can be estimated as $\frac{\theta C}{\mu}$, which is the case if the robot stays in the busy state after sending its charging request until energy depletion. We adopt this shortest remaining operation time as the rendezvous deadline for the corresponding charging task in our design, which is denoted as D_i for robot r_i. To simplify the presentation, we assume $D_i \leq T_i - T_c$ where T_c is the charging time after rendezvous, indicating the remaining energy of requesting robots is not sufficient for them to accomplish the planed travel. However, this constraint is not required for our design.

C. Construct the Travel Tree

In our design, the charging of individual robots is scheduled according to a travel tree constructed based on the robots travel profiles. Denoting the travel tree as $Tree(\mathcal{V}, \mathcal{E}, \mathcal{W})$, we discuss how to obtain the corresponding vertex set $\mathcal{V}$, edge set $\mathcal{E}$, and edge weight set $\mathcal{W}$ in the following, respectively.

1) Identify the Vertex Set: Each vertex in the travel tree has a location property and a time property, representing the feasible location and the corresponding time at which the charger may encounter the requesting robot, or the charging completion of the robot could happen. Clearly, not all the locations and the corresponding times on the robot travel profiles are feasible for these two events.

[1]When the same location appears in the trajectory multiple times at different time instances, i.e., the robot revisits certain locations, each visit of the location is treated as a unique element in its travel profile.

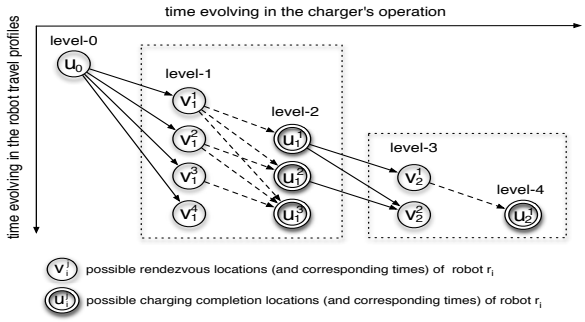

Fig. 2. Demonstration of the travel tree.

The rendezvous with robot r_i needs to be accomplished before its rendezvous deadline D_i to avoid energy depletion. Thus the locations and times after D_i in $\mathcal{P}_i$ are not feasible for the rendezvous with r_i. Furthermore, the earliest possible locations and times for the charger to rendezvous with r_i can be identified by the following best-effort charging schedule.

Best-Effort Charging Schedule: *with the best-effort charging schedule, the charger always tries to rendezvous with the to-be-charged robot as early as possible.*

Denote the charger location upon the completion of the previous charging task at time t' as a'_0 ($t' = 0$ and $a'_0 = a_0$ for the first charging task). The earliest location $a_i^{\beta_i}$ and time $\beta_i\delta$ for the charger to rendezvous with the next to-be-charged robot r_i can be identified by

$$\begin{aligned} &\min \quad \beta_i \in [t'/\delta, D_i/\delta] \\ &s.t. \qquad \parallel a'_0, a_i^{\beta_i} \parallel /v_c - \parallel a_i^{t'/\delta}, a_i^{\beta_i} \parallel /v_r <= 0. \end{aligned} \quad (1)$$

Note that if no β_i satisfies (1), it indicates the charger cannot rendezvous with r_i before its energy depletion. Thus the charger can only try to rendezvous with r_i at location $a_i^{D_i/\delta}$, where r_i is forced to stay after depleting its energy supply.

Denote $(a_i^{\beta_i}, \beta_i\delta)$ as the earliest locations and times for the charger to rendezvous with r_i. The time between the rendezvous with two consecutively charged robots is at least the charging time T_c, specifically,

$$\beta_{i+1} \geq \beta_i + T_c/\delta. \quad (2)$$

This indicates the infeasible locations and times on $\mathcal{P}_i$ increase with at least a speed of T_c/δ as i increases.

We create a vertex for each of these feasible locations and times ($\{(a_i^j, j\delta)\}\ (j \in [\beta_i, \frac{D_i}{\delta}])$), and label them as

$$\begin{aligned} Loc(v_i^k) &= a_i^{\beta_i+k-1}, \\ Time(v_i^k) &= (\beta_i + k - 1)\delta, \end{aligned}$$

where $k \in [1, 1 - \beta_i + D_i/\delta]$. A vertex set V_i containing all these vertices is created as shown in Fig. 2. Clearly,

$$|V_i| = [1 - \beta_i + D_i/\delta]^+. \quad (3)$$

The charger starts the energy replenishment of r_i after their rendezvous. With a charging time T_c, the feasible locations and times at which the charging of r_i can be completed are $\{(a_i^j, j\delta)\}\ (j \in [\beta_i + T_c/\delta, T_i/\delta])$. We create a vertex set U_i to represent these locations and times, and label them as

$$\begin{aligned} Loc(u_i^k) &= a_i^{\beta_i+\frac{T_c}{\delta}+k-1}, \\ Time(u_i^k) &= (\beta_i + T_c/\delta + k - 1)\delta, \end{aligned}$$

where $k \in [1, 1-\beta_i+(T_i-T_c)/\delta]$. Specifically,

$$|U_i| = [1-\beta_i+(T_i-T_c)/\delta]^+ . \quad (4)$$

Finally, a vertex u_0 representing the charger location at the start time is created and labeled as $< a_0, 0 >$.

The vertex set $\mathcal{V}$ of the travel tree consists of u_0, V_i, and U_i $(i = 1, 2, \cdots, m)$. The tree is rooted at u_0 and structured into $2m$ levels, where the vertices in V_i fall in its $(2i-1)$th level, and the vertices in U_i fall in its $(2i)$th level. As a result, the number of vertices in the travel tree is

$$|\mathcal{V}| = \sum_{i=1}^{m}(|V_i|+|U_i|)+1. \quad (5)$$

2) Identify the Edge Set: Next we discuss the edge set $\mathcal{E}$ in the tree. The edges can be classified into two categories. First, the edges connecting a vertex $v \in V_i$ $(i \in [1,m])$ to a vertex $u \in U_i$ reflect the charging process of the robot after its rendezvous with the charger. This type of edge is referred to as the $v \to u$ edges (the dashed arrows in Fig. 2). Second, the edges connecting a vertex $u \in U_i$ $(i \in [0, m-1])$ to a vertex $v \in V_{i+1}$ reflect the travel of the charger to rendezvous with the to-be-charged robot, which are referred to as the $u \to v$ edges (the solid arrows in Fig. 2).

For each ordered vertex pair $<u_i^j, v_{i+1}^k>$, the $u \to v$ edge $e_i^{u\to v}(j,k)$ connecting them reflects the fact that the charger travels to rendezvous with robot r_{i+1} at $Loc(v_{i+1}^k)$ from $Loc(u_i^j)$. Thus $e_i^{u\to v}(j,k)$ exists in the tree if and only if the charger can travel from $Loc(u_i^j)$ to $Loc(v_{i+1}^k)$ within the time interval of their associated time properties. Specifically,

$$e_i^{u\to v}(i,j) \in \mathcal{E} \Leftrightarrow$$
$$Time(v_{i+1}^k) - Time(u_i^j) \geq \parallel Loc(u_i^j), Loc(v_{i+1}^k) \parallel /v_c.$$

On the other hand, for ordered vertex pair $<v_i^j, u_i^k>$, the $v \to u$ edge $e_i^{v\to u}(j,k)$ connecting them means that after the rendezvous with r_i at $Loc(v_i^j)$, the energy replenishment of r_i has been accomplished at $Loc(u_i^k)$. Thus $e_i^{v\to u}(j,k)$ exists in the tree if and only if the interval between the time properties of v_i^j and u_i^k is longer than the required charging time T_c. Specifically,

$$e_i^{v\to u}(j,k) \in \mathcal{E} \quad \Leftrightarrow \quad Time(u_{i+1}^k) - Time(v_i^j) \geq T_c.$$

Following the above principles, the number of edges in the tree is upper bounded by

$$|\mathcal{E}| \quad \leq \quad \sum_{i=1}^{m}|V_i||U_i| + \sum_{i=0}^{m-1}|U_i||V_{i+1}|. \quad (6)$$

where the first summation accounts for the maximal number of $v \to u$ edges, and the second summation represents the maximal number of $u \to v$ edges.

3) Identify the Weight Set: The edge weights are assigned as the Euclidean distance between associated locations.

$$w_i^{v\to u}(j,k) \quad = \quad \parallel Loc(v_i^j), Loc(u_i^k) \parallel,$$
$$w_i^{u\to v}(j,k) \quad = \quad \parallel Loc(u_i^j), Loc(v_{i+1}^k) \parallel .$$

Overall, the complexity in constructing the travel tree is $\mathcal{O}(|\mathcal{V}|) + \mathcal{O}(|\mathcal{E}|) + \mathcal{O}(|\mathcal{E}|)$, where the three terms stand for the time required to identify the vertex set, the edge set, and the weight set, respectively. It is clear from (5) and (6) that $\mathcal{O}(|\mathcal{V}|) < \mathcal{O}(|\mathcal{E}|)$, and thus the tree construction complexity is $\mathcal{O}(|\mathcal{E}|)$.

D. Schedule based on the Travel Tree

Each path in the travel tree connecting its root u_0 and any of its leaves represents a charging schedule for the charger to carry out the m charging tasks.

Theorem 1 *No robot energy depletion results if the charger replenishes the robot energy according to a path connecting the root u_0 and any leaves in $Tree(\mathcal{V}, \mathcal{E}, \mathcal{W})$.*

This is because an edge connecting to v_i^j exists in the tree only if the charger can arrive at $Loc(v_i^j)$ before $Time(v_i^j)$, and the arrival of the charger at any vertex in U_i indicates the charging completion of r_i. We do not include the explicit proof here due to the space limit.

Based on Theorem 1, the problem of identifying a charging schedule that minimizes the charger travel distance can be transformed to find the shortest root-to-leaf path in the constructed travel tree. Thus we formulate a *Single-Source Shortest Path Problem* based on $\{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ with u_0 as the source, and adopt existing algorithms [28], [29] to find the shortest paths between u_0 and each leaf of the tree (i.e., $u_m^j \in U_m$ $(j \in [1, |U_m|])$. Then the shortest path among these paths is returned as the charging schedule of the charger.

The first step can be completed in $\mathcal{O}(|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$ time [29], and the second step needs another $\mathcal{O}(\log|U_m|)$ time. Combining with the computation time required to construct the travel tree, the time for the charger to carry out this tree-based scheduling is $\mathcal{O}(|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$.

For any $e_i^{u\to v}(j,k)$ in the returned schedule, if

$$Time(v_{i+1}^k) - Time(u_i^j) > \parallel Loc(u_i^j), Loc(v_{i+1}^k) \parallel /v_r,$$

it means the charger needs to postpone its charging service either at $Loc(u_i^j)$ (i.e., *sojourn-before-traveling*) or at $Loc(v_{i+1}^k)$ (i.e., *sojourn-after-traveling*). During the mobile charging process, new charging request may be received at any time. Thus with the *sojourn-after-traveling* approach, newly received request during the traveling may change the previously obtained schedule, and thus make the charger travel in vain. As a result, the *sojourn-before-traveling* approach is adopted in our design to reduce the unnecessary travel of the charger.

Note that the tree-based scheduling needs to be executed every time when 1) the travel profiles of requesting robots are updated, or 2) a new charging request is received. Furthermore, two events, if happen, indicate the robot energy depletion is inevitable: 1) if any of the vertex set V_i or U_i is empty, or 2) if no path from the root to any of its leaves exists in the tree. This in turn implies the remaining energy threshold θ is set too small, and thus the time left for the charger to replenish the robot energy supply before its depletion is too short. We will present guidance on identifying a proper θ in Section IV.

E. Performance Analysis

With the tree-based schedule, the mobile charger performs the charging tasks of sensor nodes according to their arrival

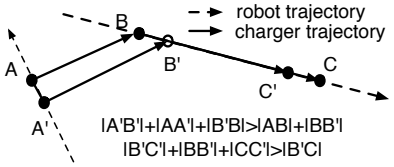

Fig. 3. The tree-based schedule achieves a near-optimal travel distance: $\overline{ABC}$ is the travel trajectory for a specific charging task returned by the tree-based charging schedule, where A is the location from which this charging task is started, B is the rendezvous location with the to-be-charged robot, and C is the location at which the charging of the robot is accomplished; $\overline{A'B'C'}$ is the trajectory returned by the optimal schedule.

sequence $\{r_1, r_2, \cdots, r_m\}$, and denote l as the obtained charger's travel distance. With this robots sequence, all the online schedules can be classified into two categories: those charge the robots with the same sequence as the tree-based schedule and those with different sequences. Next we present two theorems on the performance of the tree-based charging schedule with regard to these two schedule categories, respectively.

1) Performance w.r.t Schedules with Identical Sequence: Denote l^* as the shortest achievable travel distance with any online schedules that carries out the charging tasks with the arrival sequence. For example, when the charging requests intensity in the network is light, in most cases, the charger will carry out the charging tasks according to their arrival sequence no matter which online schedule discipline is adopted. We have the following theorem stating that the charger travel distance with the tree-based schedule l approaches l^* as the discretized time slot δ decreases.

Theorem 2 $l < l^* + 4m\delta v_r$.

In the tree-based schedule, the locations where the charger starts to rendezvous with the robots, their rendezvous locations, and the locations where the charging of the robots are accomplished, are all discretized locations on the robot travel profiles. Thus the distances between these locations and the corresponding optimal locations are upper bounded by the discretization granularity δv_r. In Fig. 3, $\overline{ABC}$ and $\overline{A'B'C'}$ are the travel trajectories to accomplish a specific charging task with the tree-based and the optimal schedule, respectively. We can extend $\overline{A'B'C'}$ by adding $\overline{A'A}$, $\overline{AA'}$, $\overline{B'B}$, and $\overline{BB'}$ to it, and the resultant tour $\overline{A'AA'B'BB'C'}$ is longer than $\overline{ABC}$ according to triangle inequality. Furthermore, with a discretization granularity δ, we know the length of $\overline{AA'}$ (and $\overline{BB'}$) is at most δv_r. Thus $l < l^* + 4m\delta v_r$.

2) Performance w.r.t. Any Schedules: In the following, we will show that even when compared with schedules that may charge the robots in different sequence, the tree-based schedule still achieves a charger travel distance that is within a constant factor to the optimal solutions. We emphasize that identifying such an optimal charging sequence, however, is NP-hard.[2]

We first derive the distribution of the rendezvous travel distance with the best-effort schedule in the following, which is needed to analytically compare the tree-based schedule with the optimal solutions. Observing the randomness in applications such as event detection and emergency rescue, we adopt the results in [30] to derive the robot location distribution in our problem, which is obtained based on randomly distributed travel destinations[3]. With some simple modifications, the robot location distribution in a square field of $L \times L$ m^2 can be analytically described by

$$f_{X,Y} = 36(x^2 - Lx)(y^2 - Ly)/L^6 \quad (0 \le x, y \le L), \quad (7)$$

from which the marginal distributions of the x and y coordinates can be derived as (h is utilized to emphasize that they are identically distributed)

$$f_X(h) = f_Y(h) = -6h^2/L^3 + 6h/L^2 \quad (0 \le h \le L). \quad (8)$$

This distribution is independent of the robot speed [30].

The charger travel distance to rendezvous with the robot is determined by the rendezvous location $a_r = (x_r, y_r)$ and the location $a_c = (x_c, y_c)$ from which the charger starts to travel for rendezvous. It is clear that a_r, essentially a location on the robot travel trajectory, follows the distribution in (7). Furthermore, with the mission-critical charging context, the charger travels with the robot together during the charging time, and thus the distribution of a_c can also be approximated as (7). Thus x_c, y_c, x_r, and y_r are *independently and identically distributed* conforming to (8), and the distribution of the rendezvous travel distance $f_{D_c}(d_c)$ can be derived, which is not included here due to the space limit. Specifically, $\mathbb{E}[d_c] \approx 0.41L$.

Clearly, the rendezvous travel distance with the tree-based schedule is shorter than that with the best-effort schedule

$$\mathbb{E}[l] \le m(\mathbb{E}[d_c] + v_r T_c). \quad (9)$$

It has been shown that with stationary charging targets, the asymptotically shortest travel distance is achieved when the charger always starts the charging tasks from the field center, and returns to the center afterwards if possible [32]. The optimality of the center-started charging process is even more obvious in the mobile-to-mobile charging context, because it is more likely for the robots to travel through the center area of the field [30]. The rendezvous travel distance d'_c in this center-started charging process can be derived with the same approach as in the appendix, and $\mathbb{E}[d'_c] \approx 0.21L$. Thus a lower bound on the asymptotically shortest travel distance l^{**} is

$$\mathbb{E}[l^{**}] \ge m(\mathbb{E}[d'_c] + v_r T_c). \quad (10)$$

From (9) and (10), we have the following theorem.

Theorem 3 *When the travel destinations are randomly distributed in a square deployment field, we have*

$$\frac{\mathbb{E}[l]}{\mathbb{E}[l^{**}]} \le \frac{\mathbb{E}[d_c] + v_r T_c}{\mathbb{E}[d'_c] + v_r T_c} < \frac{\mathbb{E}[d_c]}{\mathbb{E}[d'_c]} \approx 1.95.$$

Theorem 3 also indicates that the tree-based schedule approaches the optimal solutions as the robot speed or the required charging time increases.

F. Further Reduce the Complexity

The complexity in carrying out the tree-based schedule can be further reduced by identifying two special cases during the

[2]For any traveling salesman problem, we can construct a case of our problem with $\forall i, T_i = 0$ in polynomial time.

[3]Application-specific robot location distributions can be adopted without changing our design, which is possible to be obtained through a similar approach as in [31].

energy replenishment process. First, when there is only one request in the buffer (i.e., $m = 1$), the tree-based schedule regresses to a greedy approach that rendezvouses with the robot with the shortest travel distance, and thus no discretization is required. On the other hand, if there are relatively more requests in the buffer, it is not necessary for the charger to determine the complete schedule for all these charging tasks. This is because the longer the schedule lasts, the more likely a new request will be received before its completion, which can potentially change the current schedule.

IV. Implementation Guidance

Given a specific system deployment, the remaining energy threshold θ is the dominating parameter that determines whether it is possible for the tree-based schedule to return a depletion-free solution for the charger. It is clear that a small θ may leave the charger insufficient time to accomplish the charging tasks, in which case the robot energy depletion cannot be avoided. However, an excessively large θ is undesirable neither, because in this case the charger may rendezvous with the requesting robot only to find out whether it still has a sufficient energy supply. This not only unnecessarily increases the charger workload, but increases the charging latency of other requesting robots as well. Intuitively, the robot energy depletion occurs when their charging latency is too long. In this section, we identify the minimal achievable charging latency and the depletion probability with a given θ through a queue-based approach, and then propose guidance on the setting of θ based on them.

A. Identify the Shortest Charging Latency

For any online schedules that charge the robots with the same sequence, the best-effort charging schedule achieves the shortest charging latency with any given θ, which can be easily proved by contradiction. The clear queuing behavior in the mobile-to-mobile charging process promotes us to identify the shortest charging latency based on a queuing model, in which the charger acts as the *server* and the charging requests from the robots are the *clients*.

The distribution of the rendezvous travel time S_1 with the best-effort schedule can be derived as

$$f_{S_1}(t) = F'_{S_1}(t) = v_c f_{D_c}(v_c t) \quad (0 \le t \le \sqrt{2}L/v_c) \ .$$

With charging time T_c, the time to accomplish the energy replenishment of a robot follows the distribution[4]

$$f_S(t) = f_{S_1}(t - T_c) \quad (T_c \le t \le \sqrt{2}L/v_c + T_c). \qquad (11)$$

Specifically, $\mathbb{E}[S] = \mathbb{E}[S_1] + T_c$ and $\mathbb{V}[S] = \mathbb{V}[S_1]$.

Due to the fact that the location at which the charger starts the travel to rendezvous with the to-be-charged robot is also the location at which the previous charging task completed, the time to accomplish the charging tasks may not be independent. However, a *distribution-ergodic* property of this time can be identified with a similar approach as in [33]. This encourages us to adopt the distribution of the time between two charging completions as the service time in the queuing system.

After characterizing the service time, we next investigate the arrival of charging requests at the charger. Two operation states are possible for the robots: *busy* and *idle*. The distribution of the robot travel distance to carry out each exploring/monitoring task, denoted as d_r, can be obtained according to [30], with an average of approximately $0.53L$. Thus the asymptotic travel (busy) time for the robot to carry out an exploring/monitoring task is $\frac{0.53L}{v_r}$. Denote p as the probability for a new task to be assigned to a specific robot. The expected idle time of the robot is averaged at

$$\int_0^\infty (1-p)^{t-1} p dt = \frac{1}{p}.$$

Thus the steady-state probability for the robot to be in the *busy* state is

$$q \quad = \quad (0.53L/v_r)/(0.53L/v_r + 1/p).$$

Then the asymptotic time interval between two consecutive charging requests from a specific robot is

$$\mathbb{E}[I] = (1-\theta)C/(\mu q) + \mathbb{E}[R], \qquad (12)$$

where $(1-\theta)C$ is the energy consumption between two consecutive charging requests from a specific robot, and R is the robot charging latency. The first term on the right side of (12) is the time since the robot is fully charged last time to the time it requests charging again, and the second term represents the time since the request is sent out to the time the robot is charged.

Furthermore, the following observation has been reported and verified in many existing works [34], [35].

Theorem 4 *The superposition of n sub-processes resembles a Poisson process whose arrival rate is the sum of all the sub-processes, especially when n is large.*

We will further verify Theorem 4 with various n values in Section V-C. Thus the request arrival at the charger can be captured by a Poisson process with intensity

$$\lambda = \frac{n}{\mathbb{E}[I]} = \frac{n}{(1-\theta)C/(\mu q) + \mathbb{E}[R]}. \qquad (13)$$

With the Poisson arrival of requests and the service time distribution derived above, we can capture the charging process with an $M/G/1$ queuing model. By Pollaczek-Khinchine (P-K) formula [36], the expected response time of an $M/G/1$ queuing system is

$$\mathbb{E}[R] \quad = \quad \mathbb{E}[S] + \lambda(\mathbb{E}[S]^2 + \mathbb{V}[S])/(2(1 - \lambda\mathbb{E}[S])), \quad (14)$$

and λ can be obtained by substituting (14) into (13).

After deriving the service time distribution (S), characterizing the aggregated arrival process at the charger (Poisson arrival), and identifying its arrival intensity (λ), the construction of the queuing model is accomplished. The $M/G/1$ queuing system has been extensively studied, and we do not include the explicit results here due to the space limit. We will further verify the soundness of the model in Section V.

B. Identify the Energy Threshold

For robotic sensor networks, the threshold θ desires to be 1) as small as possible, which means less work load on the

[4]Note that in a more general case where the profile of the charging time S_2 has been identified, we can adopt the convolution theorem to substitute (11) as $f_S(t) = f_{S_1}(t) * f_{S_2}(t)$, where $*$ represents the convolution operation.

charger, and 2) the resultant energy depletion probability of the robots is no larger than a system requirement q, meaning the charging requests can be accomplished with probability $1-q$ before the corresponding requesting robots deplete their energy supply. The desired setting of θ can be identified based on the queuing model constructed above.

With a given θ, the robot charging latency distribution can be obtained based on the queuing model. Furthermore, the robot remaining operation time after requesting energy replenishment can be estimated as well. Thus the corresponding robot energy depletion probability can be obtained. Based on this observation, we identify the desired θ under the system requirement q through a *fixed-point iteration* approach. Starting with $\theta = 0$, we estimate the corresponding $\hat{\theta}_q$ according to the latency distribution. If $\hat{\theta}_q > q$, then θ is increased with a small step length ϵ. Repeat the process until the first (smallest) θ satisfying the requirement on q is identified and return it as the desired setting. It is possible to adopt more efficient method such as binary search to accelerate the iteration. The iteration returns a proper θ if the feasible solution exists, otherwise we need to either increase the robot energy capacity or relax the requirement on q.

The returned θ is essentially a lower bound of its optimal setting with the tree-based schedule, which reduces the charger travel distance at the cost of increased charging latency, and thus increases the depletion probability. Homogeneous robot energy consumption rates are assumed in above description. In a more general case with heterogeneous energy consumptions, we can find θ_i for each r_i by modifying (13) accordingly.

V. Evaluations

A. Simulation Setup

We consider a $1,000 \times 1,000\ m^2$ field where 20 to 200 robots are deployed, and one mobile charger is employed for their energy replenishment. The robot energy capacity is $40,000\ J$ (e.g., powered by four $1,000\ mAh$ $3.7\ V$ Lithium-Polymer batteries), and the consumption rates in busy state are random from $1\ J/s$ to $5\ J/s$ (e.g., a four-wheel driving robot with four $80\ mA$ $6\ V$ motors), with an average of $\frac{1+5}{2} = 3\ J/s$. These energy settings indicate an average operation time of $\frac{40,000}{3\times 3,600} \approx 2.78$ hours with a single charge, which is consistent with our empirical experience (Section II). The exploring/monitoring sites are randomly selected in the field. With these settings, we simulate the operation of the robots and update their remaining energy levels accordingly. The robots send charging requests to the charger when their remaining energy is below $\theta = 5\%$, which is set based on the proposed guidance as will be explained in Section V-C. We implement a linear charging model in the simulation, with which the robot charging time increases linearly with the amount of charged energy. This simplified charging model agrees with our empirical observation that a large portion of the battery capacity can be charged in a linear relationship with the charging time. Similar observations are reported in the data sheets of off-the-shelf battery products [37]. Unless otherwise specified, the worst case charging time is $T_c = 200\ s$, the travel speeds of the robots and the charger are $v_r = 0.2\ m/s$ [26] and $v_c = 5\ m/s$ [38], and the discretization granularity is $\delta = 50\ s$. Note the default setting of δ is determined based on our observations on δ's impact on the charging process, as will be explained later. The simulation is implemented with Matlab and the results are averaged with 100 runs.

B. Performance Evaluation

Figure 4 shows the average charger travel distance to accomplish one charging task, with the maximal charging time T_c varies from $50\ s$ to $300\ s$. Besides the results obtained with the best-effort charging schedule, the lower bounds of the shortest travel distance in Theorem 2 and Theorem 3 are also shown for comparison. We can see that by intelligently postponing its charging service, the charger travel distance can be reduced by 13–18% when compared with the best-effort schedule, and the improvement increases as the charging time increases. The resultant travel distance is around $1.05\times$ of the optimal schedule which carries out the charging tasks with the same sequence. Even when compared with schedules charging robots with any sequences, the tree-based schedule still achieves a travel distance that is about 1.45–$1.65\times$ of the lower bound of the optimal solution. These results show that the tree-based schedule achieves promising performance for the mobile charging process. However, we can also see that further improvement on the charging process is possible by adjusting the task sequence.

The travel distance with the robot number varying from 20 to 100 is shown in Fig. 5. The travel distance is reduced by around 16–23% with the tree-based schedule when compared with the best-effort schedule, and is around $1.05\times$ and 1.40–$1.52\times$ of the lower bounds with the same and any sequences, respectively. Another observation is that the travel distance with the tree-based schedule slightly increases with the robot numbers. This is because a larger network scale indicates a heavier charger workload and the robots need to wait longer before the charger rendezvouses with them after they send our the charging requests. As a result, the probability for the charger to rendezvous with the robot before its travel profile finishes is reduced, and thus the search space for the charger to find a shorter travel distance is reduced as well. For example, the rendezvous of more than 90% charging tasks is achieved before the robot travel profile finishes when 20 robots are deployed, which is reduced to 70% when $n = 100$.

Figure 6 shows the charger travel distance when discretizing the time into slots of 50–300 s. Two network scales of 20 and 100 are explored, respectively. Note that the travel distance with the best-effort schedule is relatively insensitive to the network scale (as in Fig. 5), and thus we just show that with $n = 100$ for clarity. The charger travel distance increases with a larger discretization granularity, and eventually approaches that of the best-effort schedule. Furthermore, we find that further reducing the discretized time from $50\ s$ cannot significantly reduce the travel distance. Specifically, our results show that the average travel distance is around $384\ m$ with a discretized time slot of $10\ s$ when $n = 100$,

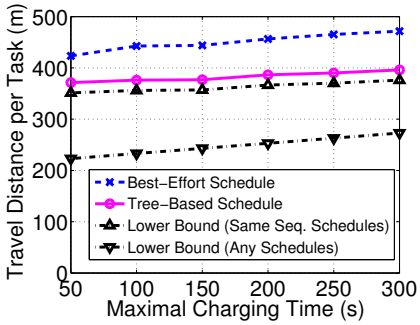

Fig. 4. Distance with T_c.

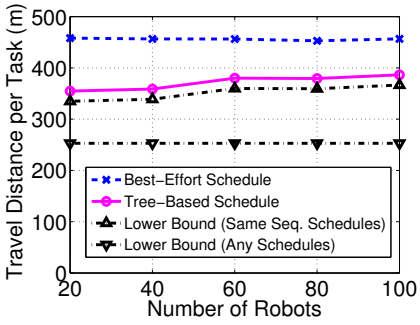

Fig. 5. Distance with n.

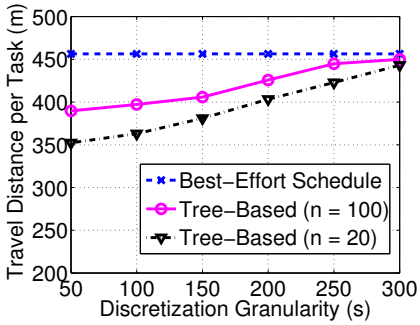

Fig. 6. Effect of δ.

TABLE I
VERIFICATION OF THE QUEUING MODEL (OUT OF 50 TRIALS)

# of Robots	20	60	100	140	180
# of Rej.	0	1	3	1	1
Arrival Corr.	0.0593	0.0276	0.0197	0.0134	0.0140
Service Corr.	0.0980	0.0965	0.0984	0.1031	0.1013

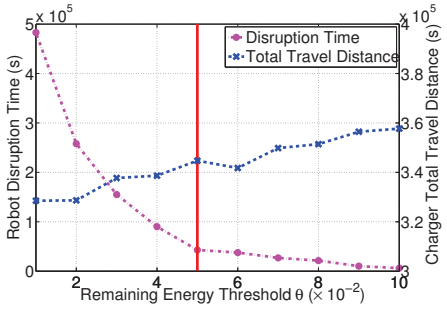

Fig. 7. Guidance on θ.

which is close to that resultant with a time slot of 50 s (389 m in Fig. 5). In our simulation, a granularity of 50 s indicates the robot travel profile is discretized into about $\frac{\mathbb{E}[d_r]}{\delta v_r} = \frac{0.53\times1{,}000}{50\times0.2} = 53$ locations, which can be easily handled with current computation power.

C. Guidance Effectiveness

Next we evaluate the effectiveness of the proposed guidance on identifying the desired θ. The guidance is based on the queue-based analysis in Section IV-A; thus before investigating the guidance performance, we first verify the soundness of the queuing model.

The Kolmogorov-Smirnov (K-S) test [39] is adopted to verify the Poisson arrival hypothesis. We carry out the tests (with a significance level of 5%) with different robot numbers (20–200), each with 50 trials. We record the number of trials that rejects the hypothesis, which is expected to be small if the hypothesis holds. The verification results are shown in the first row of Table I. We can see that even in the worst case, only $\frac{3}{50} = 6\%$ of the trials reject the hypothesis, and thus we conclude the Poisson arrival in our modeling is sound. We record the request inter-arrival time and service time, and adopt the 1-lag autocorrelation to evaluate their respective independencies. The simulation is run for 50 times with 20 to 200 robots respectively, and the absolute values of the autocorrelations are averaged as shown in the second and third rows of Table I. The correlations of both the request arrival and their service are small, which supports our modeling.

With a network scale of 100 and a maximal tolerable energy depletion probability $q = 0.0001$, the proposed guidance returns a remaining energy threshold θ of 5%. Then we simulate a network operation time period of $1,000,000$ s, with θ varies from 1% to 10%, respectively. The results are shown in Fig. 7, where the vertical line represents the result returned by the guidance. We can see that the proposed guidance can effectively identify a proper setting of θ that trades off between the robot disruption time and the charger travel distance. First, the robot disruption time will dramatically increase if the returned θ is further reduced. Second, increasing the returned θ has a limited effect in further reducing the robot disruption time, while increasing the charger travel distance.

VI. RELATED WORK

The mobile energy replenishment with stationary sensor nodes have been explored in many works [18], [19], [23], [27], [40]–[43]. However, it is fundamentally different in robotic sensor networks due to the movement of the robots.

The energy harvesting is an environmental friendly approach [8], [9]. However, the system reliability cannot be guaranteed due to the dynamics in the harvested energy [10], [11]. A more reliable approach is to deploy a stationary energy tank in the field, at which the robot energy can be replenished [7]. However, this requires the robots to travel to the tank for energy replenishment, and thus increases their load intensities. Employing mobile chargers for the energy replenishment of the robots is a reliable approach which also makes the energy replenishment easier [13]. Significant research efforts have been devoted to improving the energy replenishment performance through the coordination between the charger and robots [14]. For example, a *frugal feeding problem* was formulated and investigated in [12].

Different from existing efforts, we investigate the mission critical scenario. This is similar to classic operational research problems such as dynamic vehicle routing and job shop scheduling. However, an important difference is that, in the

mission critical scenario, the efforts (time and charger travel) required to accomplish a specific charging task is jointly determined by the locations of the charger and the robot, and thus demonstrates clear dynamic property. To the best of our knowledge, the most similar problems are 1) the aerial refueling in military operations [44], which is normally treated in the 1-dimensional space and the aerial trajectory is much more predictable and controllable; 2) the real-time hard disk scheduling [45], which falls in the mobile (i.e., the read/write head)-to-static (i.e., the place where the requested content is stored) scenario. The randomness in the 2-dimensional movement, the dual-mobility of both the charger and the robots, and the feasibility for the charger to postpone its charging service make our problem unique.

VII. Conclusions

We have presented a tree-based schedule in robotic sensor networks, which minimizes its travel distance without causing robot energy depletion. The performance of the tree-based schedule has been analytically investigated. Furthermore, we have proposed a guidance on the setting of the robot remaining energy threshold. The performances of both the tree-based schedule and the guidance have been evaluated through extensive simulation.

Acknowledgment: This research was supported in part by MultiplAtform Game Innovation Centre (MAGIC), funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority, and Singapore-MIT International Design Center IDG31000101, NSF grant CNS-1217791, the Fundamental Research Funds for the Central Universities under Grant 2013QNA5013, NSERC Canada, iTrust IGDSi1301013.

References

[1] J. M. Dolan, G. Podnar, S. B. Stancliff, E. L. Ratliff, J. Higinbotham, J. Hosler, T. Ames, J. Moisan, T. Moisan, and A. Elfes, "Harmful algal bloom characterization via the telesupervised adaptive ocean sensor fleet," in *NSTC'07*.

[2] C. Detweiler, M. Doniec, M. Jiang, M. Schwager, R. Chen, and D. Rus, "Adaptive decentralized control of underwater sensor networks for modeling underwater phenomena," in *Sensys'10*.

[3] Y. Wang, R. Tan, G. Xing, J. Wang, and X. Tan, "Accuracy-aware aquatic diffusion process profiling using robotic sensor networks," in *IPSN'12*.

[4] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *IEEE INFOCOM'13*, 2013.

[5] J. R. Department and J. Reich, "Robot-sensor networks for search and rescue," in *SSRR'06*.

[6] Y. Wang, R. Tan, G. Xing, X. Tan, J. Wang, and R. Zhou, "Spatiotemporal aquatic field reconstruction using robotic sensor swarm," in *RTSS'12*.

[7] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *IEEE ICRA'02*.

[8] T. Zhu, Y. Gu, T. He, and Z. Zhang, "eShare: a capacitor-driven energy storage and sharing network for long-term operation," in *Sensys'10*.

[9] J. Gummeson, S. Clark, K. Fu, and D. Ganesan, "On the limits of effective hybrid micro-energy harvesting on mobile CRFID sensors," in *MobiSys'10*.

[10] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *MobiSys'09*.

[11] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-Energy: a novel energy prediction model for solar and wind energy harvesting WSNs," in *MASS'12*, 2012.

[12] Y. Litus, R. T. Vaughan, and P. Zebrowski, "The frugal feeding problem: Energy efficient, multi-robot, multi-place rendezvous," in *IEEE ICRA'07*.

[13] P. Zebrowski and R. Vaughan, "Recharging robot teams: a tanker approach," in *IEEE ICAR'05*.

[14] A. Drenner, M. Janssen, and N. Papanikolopoulos, "Coordinating recharging of large scale robotic teams," in *IROS'09*.

[15] "CT3000 Charding Station," http://www.verdek.com.

[16] A. Noda and H. Shinoda, "Selective wireless power transmission through high-Q at waveguide-ring resonator on 2-D waveguide sheet," *Trans. on Micro. Theo. and Tech.*, vol. 59, no. 8, pp. 2158–2167, 2011.

[17] "Roomba," http://www.irobot.com/.

[18] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "Prolonging sensor network lifetime through wireless charging," in *RTSS'10*.

[19] Z. Li, Y. Peng, W. Zhang, and D. Qiao, "J-RoC: a joint routing and charging scheme to prolong sensor network lifetime," in *ICNP'11*.

[20] M. Khouzani, S. Eshghi, S. Sarkar, S. S. Venkatesh, and N. B. Shroff, "Optimal energy-aware epidemic routing in DTNs," in *MobiHoc'12*.

[21] K. Han, L. Xiang, J. Luo, and Y. Liu, "Minimum-energy connected coverage in wireless sensor networks with omni-directional and directional features," in *MobiHoc'12*.

[22] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *MobiHoc'08*.

[23] L. He, Y. Gu, J. Pan, and T. Zhu, "On-demand charging in wireless sensor networks: Theories and applications," in *MASS'13*, 2013.

[24] Y. Shi, L. Xie, Y. T. Hou, and H. D. Sherali, "On renewable sensor networks with wireless energy transfer," in *INFOCOM'11*.

[25] "Pioneer 3-AT," http://www.mobilerobots.com/researchrobots/p3at.aspx.

[26] "Scout Robot," http://distrob.cs.umn.edu/scout.php.

[27] L. He, Y. Gu, and T. He, "Poster abstract: Energy synchronized charging in sensor networks," in *SenSys'12*.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms (3rd ed.)," *The MIT Press*, 2009.

[29] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Jour. of the Asso. for Comp. Mach.*, vol. 34, no. 3, pp. 596–615, 1987.

[30] C. B. H. Hartenstein and X. Perez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[31] E. Hyytia, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *Trans. on Mobi. Comp.*, vol. 5, no. 6, pp. 680 – 694, 2006.

[32] J. Bertsimas and G. V. Ryzins, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," *Operations Research*, vol. 39, pp. 601–615, 1991.

[33] 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 Transactions on Mobile Computing*, vol. 99, no. PrePrints, p. 1, 2013.

[34] S. L. Albin, "On Poisson approximations for superposition arrival processes in queues," *Management Science*, vol. 28, no. 2, pp. 126–137, Feb. 1982.

[35] C. Y. T. Lam and J. P. Lehoczky, "Superposition of renewal processes," *Advances in Applied Probability*, vol. 23, no. 1, pp. pp. 64–85, 1991.

[36] D. Gross, "Fundamentals of Queueing Theory (4th ed.)," *New Jersey: John Wiley & Sons*, p. 232, 2008.

[37] "NCR18500," http://www.panasonic.com.

[38] "AlphaDog," http://www.pcmag.com/article2/0,2817,2409628,00.asp.

[39] "K-S test," http://www.physics.csbsju.edu/stats/KS-test.html.

[40] S. Guo, C. Wang, and Y. Yang, "Mobile data gathering with wireless energy replenishment in rechargeable sensor networks," in *INFOCOM'13*, 2013.

[41] P. Cheng, S. He, F. Jiang, Y. Gu, and J. Chen, "Optimal scheduling for quality of monitoring in wireless rechargeable sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 3072–3084, 2013.

[42] L. Fu, P. Cheng, Y. Gu, J. Chen, and T. He, "Minimizing charging delay in wireless rechargeable sensor networks," in *IEEE INFOCOM'13*, 2013.

[43] S. He, J. Chen, F. Jiang, D. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *EEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931–1942, 2013.

[44] R. K. Nangia, "Operations and aircraft design towards greener civil aviation using air-to-air refuelling," *The Aeronautical Journal*, no. 3088, pp. 705–721, 2011.

[45] T. J. Teorey and T. B. Pinkerton, "A comparative analysis of disk scheduling policies," *Comm. ACM*, vol. 15, no. 3, pp. 177–184, 1972.