On Optimal Clustering in Mobile Wireless Sensor Networks Under Uncertainty

Guvenc Degirmenci[1] Jeffrey P. Kharoufeh[2] Oleg A. Prokopyev[3]
Department of Industrial Engineering
University of Pittsburgh
1048 Benedum Hall
3700 O'Hara Street
Pittsburgh, PA 15261 USA

Accepted Version

To appear in
Military Operations Research

Abstract

Wireless sensor networks are comprised of geographically distributed sensors that collect and convey time-critical information without human intervention, making them particularly useful for military applications; however, network performance may be limited by the typically small energy supplies of the sensors. One strategy for dealing with limited energy storage in such networks is to allow sensor nodes to aggregate sensed data at particular nodes known as cluster heads. We examine the problem of optimally locating and relocating cluster heads over a finite time horizon. The objective is to simultaneously maximize the demand coverage and minimize the costs of relocating cluster heads. Our main contribution is to consider explicitly the random evolution of mobile sensor nodes, which requires periodic updating of their precise locations. Computational experiments illustrate the usefulness of such dynamic information updates to help mitigate the uncertainty in sensor node locations and ensure reliable demand coverage by cluster heads.

INTRODUCTION

A wireless sensor network (WSN) is a collection of small, low-cost sensing devices (called sensor nodes), linked via a wireless communication medium, that behave cooperatively to perform network tasks in a distributed manner. WSNs are emerging in such diverse applications as ecological and environmental monitoring, structural health monitoring, industrial process control and military surveillance. They are attractive in a variety of applications because they can sense and convey critical information about objects and their surroundings without human intervention. For example, in ecological monitoring applications, it may be desirable to aggregate temperature and humidity readings from geographically dispersed sensors at a central processing unit. In this paper, we are particularly motivated by military communications over a wireless ad hoc network in which multiple dispersed mobile sensors gather information about their surroundings and generate data streams that can be fused for a variety of purposes (e.g., to assess the current battlefield scenario,

[1]Email: gdegirmenci@gmail.com.
[2]Corresponding author. Email: jkharouf@pitt.edu; Ph: (412) 624-9832; Fax: (412) 624-9831.
[3]Email: droleg@pitt.edu.

to identify and track moving targets, or to provide general threat conditions within a region). Wireless sensor networks in a time-dynamic military theater are prone to failure due to disruptions caused by adversaries or the movement of sensors beyond the communication range of their base stations. Additionally, because sensor nodes are typically small in size, they have very limited energy reserves, local memory, transmission range and computational capabilities. Therefore, it is imperative to utilize network resources prudently in order to maximize sensor node lifetimes and the likelihood that the network remains connected.

One strategy for dealing with limited energy storage in WSNs is to allow the sensor nodes to aggregate sensed data at a particular node (or set of nodes) in the network known as a *cluster head*. A cluster head typically has a larger energy supply and transmission range than other nodes in the network (Younis, Krunz, & Ramasubramanian, 2006) and is responsible for aggregating and disseminating data received from sensors within its own cluster. A *cluster* is a collection of sensor nodes within the transmission range of a cluster head. Two important design issues emerge when cluster heads are used: *(i)* Where should cluster heads be located in a particular region; and *(ii)* How should the sensor nodes be assigned to cluster heads, given that the positions of the sensors evolve dynamically (and possibly randomly) over time? These issues are complicated by unreliable communications due to a harsh operating environment or attacks on the network by an adversary (e.g., in a military theater).

The problem of clustering sensor nodes in a WSN has been analyzed from a variety of perspectives including load balancing, fault-tolerance, connectivity, cluster count and maximizing network lifetime, to name only a few. Some useful survey papers related to clustering in WSNs include Liu and Shi (Liu & Shi, 2012), Kumar et al. (Kumar, Jain, & Tiwari, 2011), Younis et al. (Younis et al., 2006) and Abbasi and Younis (Abbasi & Younis, 2007). Liu and Shi surveyed popular clustering algorithms and categorized them into three groups: cluster head election algorithms, cluster formation and data transmission (Liu & Shi, 2012). Low Energy Adaptive Clustering Hierarchy (LEACH) is one of the most popular clustering algorithms that uses a distributed algorithm in which cluster heads are selected and rotated randomly to balance energy expenditure among the network's sensors (Heinzelman, Chandrakasan, & Balakrishnan, 2000). A node becomes a cluster head with a probability assigned by the algorithm. LEACH assumes that all sensor locations are known and the number of clusters in the network is fixed *a priori* in the algorithm. In the Hybrid Energy Efficient Distributed (HEED) clustering algorithm (Younis & Fahmy, 2004) and Time Delay Based Clustering (TDC) algorithm (Zhong, Wang, Xu, Yu, & Xu, 2007), cluster head selection is based on residual energy, and explicit sensor location information is not required; however, the optimal selection of cluster heads is not guaranteed. In HEED (Younis & Fahmy, 2004), intra-cluster communication costs – the energy expenditure due to communication from a sensor to its cluster head – are not considered, whereas the Distributed, Weight-based Energy-efficient Hierarchical Clustering (DWEHC) protocol does consider these intra-communication costs (Ding, Holliday, & Celik, 2005). Yu et al. presented a clustering algorithm for large-scale networks that minimizes both intra- and inter-cluster communications costs (Yu, Leung, & Malvankar, 2007). Xia and Vlajic (Xia & Vlajic, 2007) and Manisekaran et al. (Manisekaran, Venkatesan, & Deivanai, 2011) developed algorithms in which the similarity of sensor readings is used as the main clustering criterion to minimize the network communications. Dimoskas et al. considered the significance of a sensor with respect to its contribution in relaying messages as a metric for clustering (Dimokas, Katsaros, & Manolopoulos, 2010). Koucheryavy and Salim developed a multiple-criteria metric comprised of connectivity, coverage, mobility and residual energy; their distributed algorithm uses predicted values of this combined metric (Koucheryavy & Salim, 2010). Each of these techniques can be viewed as distributed algorithms that make decisions based on *local* observations. However, they have extensive data collection and storage requirements (Ci, Guizani, & Sharif, 2007).

By contrast, centralized algorithms have been proposed to reduce network energy expenditure. LEACH-C (Heinzelman, Chandrakasan, & Balakrishnan, 2002) is a centralized version of LEACH wherein the rotation of cluster heads is controlled by a base station. Based on computational results, LEACH-C is superior to LEACH in reducing energy consumption. Ci et al. introduced data mining into the network design problem by inferring sensor locations and network topology using only observations of the remaining energy at each of the nodes without localization data (Ci et al., 2007). Khan et al. proposed Multiple Parameter-based Clustering (MPC) which makes clustering decisions based on the residual energy of sensors, proximity to the base station, and latency of data to the base station (Khan, Madani, Hayat, & Khan, 2012). Gupta et al. proposed a fuzzy-logic based clustering algorithm that considers the residual energy, the number of neighboring nodes and centrality of the nodes (Gupta, Riordan, & Sampalli, 2005).

Relevant to our work are models that apply both exact and heuristic optimization techniques to the optimal clustering problem in WSNs. Slama et al. developed an optimization model that maximizes network lifetime by balancing energy expenditure over the sensors' activities (Slama, Ghedira, Jouaber, & Afifi, 2007). Islam et al. proposed heuristic methods for the same problem (Islam, Hyder, Kabir, & Naznin, 2010). Krivitski et al. proposed a facility location-based heuristic algorithm for sensor networks where resources can be placed in any of the k out of m possible locations (Krivitski, Schuster, & Wolff, 2005). Furuta et al. formulated the WSN clustering problem as an uncapacitated facility location problem and incorporated the residual energy of sensor nodes (Furuta, Sasaki, Ishizaki, Suzuki, & Miyazawa, 2009). Their objective is to extend the network lifetime by finding the optimal number of cluster heads and simultaneously selecting the cluster head candidates. Youssef et al. proposed a heuristic that considers cluster overlap in addition to connectivity and coverage (Youssef, Youssef, & Younis, 2009). Their heuristic provides the set of cluster heads such that every node in the network is within some distance k from a cluster head. Aioffi et al. proposed algorithms to minimize message delivery latency by considering topology constraints that reduce energy consumption (Aioffi, Valle, Mateus, & Cunha, 2011). Shanbehzadeh et al. developed a genetic algorithm and particle swarm-based heuristic algorithm to determine the number of clusters, select the cluster heads and cluster the members (Shanbehzadeh, Mehrjoo, & Sarrafzadeh, 2011). Sevgi and Kocyigit developed a heuristic to determine the size of clusters and initial energy level of sensors to maintain network lifetime and coverage requirements (Sevgi & Kocyigit, 2009).

Perhaps most relevant to our model is the one described by Patel et al. who considered the optimal clustering of nodes in an ad hoc network with mobile nodes and unreliable communication links (Patel, Batta, & Nagi, 2005). They proposed a mixed integer linear programming (MILP) model to maximize the expected data coverage, less the cluster head reassignment costs. They developed a column-generation heuristic to solve the problem and suggested several interesting extensions. Their models assume that sensor locations (or at least distances to cluster heads) are known *a priori* and do not consider dynamic and stochastic evolution sensor movements within the region.

The primary objective of this paper is to formulate and solve optimization problems that seek to maximize the demand coverage and minimize the costs of locating, and relocating, cluster heads in a WSN with mobile nodes and unreliable links. Specifically, we will determine the cluster head locations and assignment of sensor nodes to particular cluster heads for each period in a finite planning horizon. As in (Patel et al., 2005), cluster heads are relocated over time to ensure that each sensor node can send information to a cluster head in a single hop. However, due to node mobility and unreliability of the links, we also consider the optimal timing of sensor location updates to maintain connectivity of the network and improve demand coverage. Our model can be viewed as an extension of the model in (Patel et al., 2005) in that we allow the positions of sensor nodes to evolve randomly over time, and we explicitly determine the best time(s) to update sensor location information. Our computational experiments illustrate the usefulness of such dynamic information updates, which allow one to mitigate uncertainty in sensor node locations and ensure reliable demand coverage by cluster heads.

The remainder of the paper is organized as follows. The next section introduces essential notation, provides the main problem formulation for optimally locating and relocating cluster heads, and describes connections to prior work. Subsequently, we formulate the problem of optimally timing information updates on sensor node locations. Next, we highlight the advantages of this updating by way of numerical examples illustrating the improvements in coverage that can be achieved by dynamically (and optimally) updating sensor location information. Finally, we provide some concluding remarks and future research directions.

PROBLEM FORMULATION

Consider a multi-hop WSN represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{1, \ldots, N\}$ is the set of nodes, N is the number of sensor nodes in the network, and $\mathcal{A}$ is the set of feasible arcs in the sensor network. Each sensor node is assumed to use a fixed transmission range r (meters). For $\ell, m \in \mathcal{N}$, an arc (ℓ, m) is an element of $\mathcal{A}$ if and only if nodes ℓ and m are within transmission range of one another. Initially, the nodes are randomly distributed in a region $R \subset \mathbb{R}^2$; however, the nodes are mobile, and their locations evolve according to a continuous-time, continuous-state stochastic process (described in greater detail later). For the sake of clarity, we assume that R is a square sensor field, but the same model is applicable even if R is not square or if $R \subseteq \mathbb{R}^3$. There are finitely many points in R that are candidate locations for cluster heads. Assets placed at these locations are not elements of $\mathcal{N}$; rather, they are temporary base stations for aggregating and disseminating data from the sensor nodes in $\mathcal{N}$. Let $\mathcal{S} = \{1, 2, \ldots, K\}$ be the set of labels for candidate cluster head locations. At most n ($n \leq K$) of the K candidate locations can be selected to host cluster heads. Figure 1 graphically depicts this scenario where points marked by $\otimes$ represent candidate cluster head locations and the black dots represent the mobile sensor nodes.

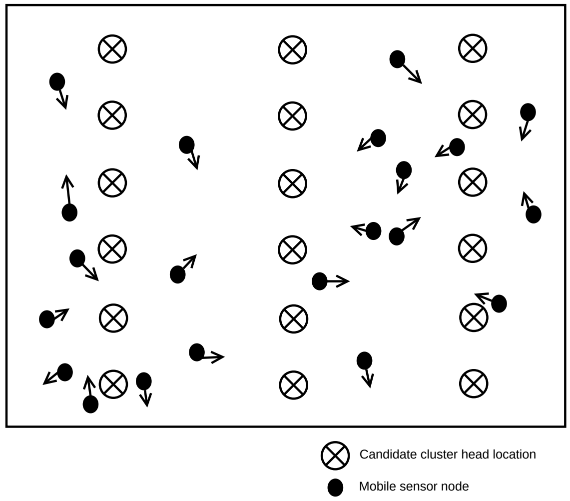


Figure 1: Graphical depiction of a WSN with mobile nodes and cluster head locations.

The planning horizon is represented by a finite set $\mathcal{T} = \{1, \ldots, T\}$ of equal-duration time periods indexed by $t \in \{1, \ldots, T\}$. During each period $t \in \mathcal{T}$, node $j \in \mathcal{N}$ generates a deterministic demand d_j that must be satisfied (or covered) by its cluster head. This demand might, for example, be the number of temperature readings reported to the cluster head during the period. The following notation will be used in the main model:

Parameters:

- d_j: the per period demand generated by sensor $j \in \mathcal{N}$;
- p: the probability of link failure between any cluster head and sensor $(0 < p < 1)$;
- C: the unit cost of placing a single cluster head (or changeover cost);
- Z_{ijt}: an indicator random variable for the proximity of cluster head i to sensor j in period t,

$$Z_{ijt} = \begin{cases} 1, & \text{if cluster head location } i \text{ is within range of sensor } j \text{ in period } t, \\ 0, & \text{otherwise.} \end{cases}$$

In the main model, the objective is to locate and relocate cluster heads, and to assign sensors to cluster heads during each period of the planning horizon. Therefore, the decisions variables are as follows:

Decision Variables:

- x_{it}: a binary variable for cluster head assignment, where

$$x_{it} = \begin{cases} 1, & \text{if a cluster head is placed at location } i \text{ in period } t, \\ 0, & \text{otherwise;} \end{cases}$$

- υ_{jkt}: a binary variable for the coverage of node j by at least k cluster heads, where

$$\upsilon_{jkt} = \begin{cases} 1, & \text{if sensor } j \text{ is covered by at least } k \text{ cluster heads in period } t, \\ 0, & \text{otherwise;} \end{cases}$$

- w_{it}: a binary variable for the status of relocation of cluster head i at period t, where

$$w_{it} = \begin{cases} 1, & \text{if a cluster head is located at } i \text{ in period } t-1 \text{ and not in period } t, \\ 1, & \text{if a cluster head is located at } i \text{ in period } t \text{ and not in period } t-1, \\ 0, & \text{otherwise.} \end{cases}$$

The following mathematical programming formulation (problem $\mathbf{P}_1$) can be viewed as an extension of the model presented by Patel et al. (Patel et al., 2005), where rather than specifying a deterministic constraint for the connectivity of sensors and cluster heads, we use a probabilistic constraint, namely (2), that allows us to bound the likelihood that the connectivity requirement is violated.

$$
\begin{aligned}
(\mathbf{P}_1) \quad \max \quad & \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}}\sum_{k=1}^{K}(1-p)p^{k-1}d_j\upsilon_{jkt} - C\sum_{i\in\mathcal{S}}\sum_{t\in\mathcal{T}} w_{it} && && (1)\\
\text{s.t.} \quad & \mathbb{P}\left(\sum_{k=1}^{K}\upsilon_{jkt} - \sum_{i\in\mathcal{S}} Z_{ijt}\, x_{it} > \psi\right) \le \kappa && \forall j\in\mathcal{N},\, t\in\mathcal{T}, && (2)\\
& \sum_{i\in\mathcal{S}} x_{it} \le n && \forall t\in\mathcal{T}, && (3)\\
& w_{it} \ge x_{i,t-1} - x_{it} && \forall i\in\mathcal{S},\, t\in\mathcal{T}\setminus\{1\}, && (4)\\
& w_{it} \ge x_{it} - x_{i,t-1} && \forall i\in\mathcal{S},\, t\in\mathcal{T}\setminus\{1\}, && (5)\\
& x_{it}\in\{0,1\},\ w_{it}\in\{0,1\} && \forall i\in\mathcal{S},\, t\in\mathcal{T}, && (6)\\
& \upsilon_{jkt}\in\{0,1\} && \forall j\in\mathcal{N},\, k\in\mathcal{S},\, t\in\mathcal{T}. && (7)
\end{aligned}
$$

The objective function (1) represents the expected demand covered by cluster heads minus the total relocation costs. The first term, $\sum_{k=1}^{K}(1-p)p^{k-1}d_j\upsilon_{jkt}$, corresponds to the expected demand coverage of sensor j in period t, considering link reliability. In the second term, the cost C can be viewed as a weighting parameter between the expected demand coverage and total relocation costs. This relocation cost is associated with moving from one location and moving to a new location. Therefore, a fixed cost of $2C$ is incurred if a cluster head is moved. Although the units of the objective function are essentially meaningless, the numerical values are useful for comparing the quality of different solutions. Constraints (2) ensure that if sensor j is covered by k' cluster heads at time t, then each of the variables $\upsilon_{j1t}, \upsilon_{j2t}, \ldots, \upsilon_{jk't}$ are assigned a value of 1 since the objective function contains the term υ_{jkt}. Note in constraints (2) that Z_{ijt} is a random variable describing the connectivity of cluster head i and sensor j at time t. Here, ψ ($\psi > 0$) can be viewed as a small measure of infeasibility, and κ ($0 \le \kappa \le 1$) is a user-specified probability that bounds the likelihood of violating the chance constraint. Larger values of κ serve to relax the problem but may not yield high-quality solutions. On the other hand, when $\kappa = 0$, there is no chance of violating the constraint, yielding the most conservative solution. Furthermore, if $\kappa = 0$ and the values of Z_{ijt} are deterministic and known exactly for each $i \in \mathcal{S}$, $j \in \mathcal{N}$ and $t \in \mathcal{T}$, then (2) can be equivalently re-written as

$$\sum_{k=1}^{K} \upsilon_{jkt} - \sum_{i \in \mathcal{S}} Z_{ijt}\, x_{it} \leq 0, \qquad j \in \mathcal{N},\, t \in \mathcal{T},$$

in which case $\mathbf{P}_1$ reduces to the optimization model examined by Patel et al. (Patel et al., 2005). Constraints (3) ensure that the maximum number of cluster heads to be located does not exceed n during any period. Constraints (4) and (5) determine the cluster head relocations and force w_{it} to equal 1 if there is a change in cluster head location i in terms of cluster head assignments at time t. The admissible ranges of variables are given by constraints (6) and (7).

If the probability distribution of Z_{ijt} is known, it can be used to write the chance constraints (2) in a deterministic form that depends on the parameter κ. However, even if the distribution function of Z_{ijt} were known, computing the left-hand side of (2) requires the distribution of the random variable $\sum_{i \in \mathcal{S}} Z_{ijt}\, x_{it}$, which, in general, is not easily attainable. To circumvent this complication, we next present an alternative optimization model. Define the following additional variables and parameters:

- z_{ijt}: a binary variable for the connectivity of cluster head i and sensor j at time t, where

$$z_{ijt} = \begin{cases} 1, & \text{if a cluster head at location } i \text{ is within range of sensor } j \text{ at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

- y_{jt}: a binary variable for the coverage of sensor j at time t, where

$$y_{jt} = \begin{cases} 1, & \text{if sensor } j \text{ is covered by at least 1 cluster head with at least probability } \kappa \text{ at } t, \\ 0, & \text{otherwise.} \end{cases}$$

- $\mathcal{S}_{jt} \subseteq \mathcal{S}$: the subset of cluster heads within range of sensor j at time t, where

$$\mathcal{S}_{jt} = \{ i \in \mathcal{S} : \mathbb{P}(Z_{ijt} = 1) > \kappa \};$$

- q_{jt}: the probability that sensor j is covered by at least one cluster head and the link between the sensor and cluster head is operational.

Let Q_{ijt} be a random variable indicating the status of the link between sensor j and cluster head i at time t. That is, $Q_{ijt} = 1$ if the link between cluster head i and sensor j is failed and 0, otherwise. Note that $\mathbb{P}(Q_{ijt} = 1) = p$. Then:

Proposition 1 *If the random variables $\{Q_{ijt} : i \in \mathcal{S},\, j \in \mathcal{N},\, t \in \mathcal{T}\}$ are mutually independent, then*

$$q_{jt} = 1 - \prod_{i \in \mathcal{S}_{jt}} \left[\mathbb{P}(Z_{ijt} = 1) p + \mathbb{P}(Z_{ijt} = 0) \right]$$

for each $j \in \mathcal{N}$ and $t \in \mathcal{T}$.

Next, let $\mathbb{1}_{\{\chi\}}$ be an indicator function that assumes the value 1 if condition χ holds and 0, otherwise. Then the probabilistic constraints (2) are replaced by

$$z_{ijt} - \mathbb{1}_{\{\mathbb{P}(Z_{ijt}=1)>\kappa\}}\, x_{it} \le 0,$$

and the revised model is given by:

$$
\begin{aligned}
(\mathbf{P}_2) \quad & \max \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}} q_{jt} d_j y_{jt} - C \sum_{i\in\mathcal{S}}\sum_{t\in\mathcal{T}} w_{it} && && (8)\\
& \text{s.t. } (3) - (6), && && \\
& z_{ijt} - \mathbb{1}_{\{\mathbb{P}(Z_{ijt}=1)>\kappa\}}\, x_{it} \le 0 && \forall\, i \in \mathcal{S},\, j \in \mathcal{N},\, t \in \mathcal{T}, && (9)\\
& y_{jt} - \sum_{i\in\mathcal{S}} z_{ijt} \le 0 && \forall\, j \in \mathcal{N},\, t \in \mathcal{T}, && (10)\\
& y_{jt} \in \{0,1\},\ \ z_{ijt} \in \{0,1\} && \forall\, i \in \mathcal{S},\, j \in \mathcal{N},\, t \in \mathcal{T}. && (11)
\end{aligned}
$$

As in problem $\mathbf{P}_1$, the objective function (8) represents the expected demand covered by cluster heads minus the relocation costs. Constraints (9) and (10) ensure that $y_{jt} = 0$ if no cluster head is located within the transmission range of sensor j with at least probability κ in period t. It is worth mentioning that the column generation algorithm proposed by Patel et al. (Patel et al., 2005) can be applied to solve problem $\mathbf{P}_2$ when $\mathbb{1}_{\{\mathbb{P}(Z_{ijt}=1)>\kappa\}}$ is known.

So far, we have assumed that the planning horizon is known *a priori*, and only the *initial* positions of the sensors are known. However, to improve WSN performance, the mobile sensors can broadcast their locations periodically. This updating can be costly because the additional network traffic may, for example, increase energy expenditure (Wang et al., 2007), thereby reducing the useful lifetime of sensor nodes. The next section describes an extension of $\mathbf{P}_2$ to determine the optimal time to *next* update the sensor locations in order to maximize the expected demand coverage and minimize the location/relocation costs over the planning horizon.

OPTIMALLY TIMING LOCATION UPDATES

The optimal solutions of problems $\mathbf{P}_1$ and $\mathbf{P}_2$ yield myopic policies in that they maximize the demand coverage and minimize location/relocation costs using only the initial sensor locations. However, such information may not be adequate for decision making in each of the subsequent periods of the planning horizon. In this section, we extend model $\mathbf{P}_2$ to determine the next time to update the sensor locations. That is, at the beginning of period $t = 1$, the model will determine not only cluster head and sensor assignments, but also the optimal time t^* at which to update sensor location information. The problem can be solved sequentially starting with the updated sensor coordinates at time t^* until the end of the planning horizon. These optimal times lead to improved demand coverage and lower relocation costs.

Define s_t as a binary variable describing the first time the node locations are updated after time 0, that is,

$$s_t = \begin{cases} 1, & \text{if sensor locations are updated at time } t, \\ 0, & \text{otherwise,} \end{cases}$$

and let C' be the unit cost of updating the sensor locations. The new formulation is given by:

$$(\mathbf{P}_3) \quad \max \left(\sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{N}} q_{jt} d_j y_{jt} - C \sum_{i\in\mathcal{S}} \sum_{t\in\mathcal{T}} w_{it} - C' \right) \cdot \sum_{t\in\mathcal{T}} \frac{s_t}{t} \tag{12}$$

$$\text{s.t.} \quad (3) - (6),\ (9),\ (10),\ (11)$$

$$y_{jt} + \sum_{t'\leq t} s_{t'} \leq 1 \qquad \forall\, j \in \mathcal{N},\, t \in \mathcal{T}, \tag{13}$$

$$\sum_{t\in\mathcal{T}} s_t = 1 \tag{14}$$

$$s_t \in \{0,1\} \qquad \forall\, t \in \mathcal{T}. \tag{15}$$

The objective function (12) is identical to (8) except that here we maximize the average expected demand covered by cluster heads minus the relocation and information updating costs from the prior updating time until the next updating time. That is, the term $\sum_{t\in\mathcal{T}} s_t/t$ ensures that the objective function value is averaged over the time between two consecutive sensor location updates. Constraints (13) ensure that sensor assignments are made for time periods prior to the next optimal updating time such that $y_{jt} = 0$ for $t \geq t^*$, where t^* is the next update time. These constraints are required because sensor locations are updated at time t^*, and $\mathbf{P}_3$ can be solved with the updated information for time periods $t \in \mathcal{T} \setminus \{1, 2, \dots, t^* - 1\}$. Constraint (14) ensures that only the next update time is determined. Next, we summarize the solution procedure to elucidate how relocation decisions are made for each period $t \in \mathcal{T}$.

Step 0: Initialize or update q_{jt} for $j \in \mathcal{N}$, $t \in \mathcal{T}$.

Step 1: Solve problem $\mathbf{P}_3$ to obtain cluster head locations for $t \in \mathcal{T}$ and the update time t^*.

Step 2: At time t^*, update the time horizon so that $\mathcal{T} := \mathcal{T} \setminus \{1, 2, \dots, t^* - 1\}$.

Step 3: If $|\mathcal{T}| > 1$, go to **Step 0**; else end.

Solving $\mathbf{P}_3$ is nontrivial due to the nonlinearity of the objective function (12); however, this complication can be avoided by linearizing the model using a standard big-M approach (Glover, 1975). Doing so facilitates solving the problem with a commercial MILP solver, such as CPLEX. To facilitate linearization, we introduce the following variables:

- u_{jt}: a continuous variable describing time-averaged coverage prior to the next update,

$$u_{jt} = y_{jt} \sum_{t'\in\mathcal{T}} \frac{s_{t'}}{t'} \qquad \forall\, j \in \mathcal{N},\, t \in \mathcal{T}; \tag{16}$$

- v_{it}: a continuous variable describing time-averaged relocation prior to the next update,

$$v_{it} = w_{it} \sum_{t'\in\mathcal{T}} \frac{s_{t'}}{t'} \qquad \forall\, i \in \mathcal{S},\, t \in \mathcal{T}. \tag{17}$$

Using these variables, we next present a MILP to determine the optimal cluster head locations and the information updating times over the planning horizon $\mathcal{T}$:

$$(\mathbf{P}_4) \quad \max \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}} q_{jt}d_j u_{jt} - C\sum_{i\in\mathcal{S}}\sum_{t\in\mathcal{T}} v_{it} - C'\sum_{t\in\mathcal{T}} \frac{s_t}{t} \tag{18}$$

$$\text{s.t.} \quad (3)-(6),\ (9),\ (10),\ (11),\ (13)-(15),$$

$$u_{jt} - \sum_{t'\in\mathcal{T}} \frac{s_{t'}}{t'} \le 0 \qquad \forall\, j\in\mathcal{N},\ t\in\mathcal{T}, \tag{19}$$

$$u_{jt} - y_{jt} \le 0 \qquad \forall\, j\in\mathcal{N},\ t\in\mathcal{T}, \tag{20}$$

$$w_{it} - v_{it} + \sum_{t'\in\mathcal{T}} \frac{s_{t'}}{t'} \le 1 \qquad \forall\, i\in\mathcal{S},\ t\in\mathcal{T}, \tag{21}$$

$$v_{it} \ge 0,\ u_{jt} \ge 0 \qquad \forall\, i\in\mathcal{S},\ j\in\mathcal{N},\ t\in\mathcal{T}. \tag{22}$$

The objective function (18) of $\mathbf{P}_4$ is identical to that of problem $\mathbf{P}_3$ except that it has been linearized. Constraints (19) through (21) ensure that u_{jt} and v_{it} are assigned their defined values via (16) and (17), and their non-negativity is established by inequalities (22). Most significantly, the linearized model $\mathbf{P}_4$ is amenable to solution by a commercial solver (*ILOG CPLEX*, 2015). The next section highlights the advantages of dynamically updating information on sensor locations by illustrating the case when sensor positions evolve independently according to a two-dimensional Brownian motion process with drift.

COMPUTATIONAL RESULTS

In this section, we illustrate the advantages of updating sensor locations over time by way of two numerical examples. To this end, it is assumed that each sensor's movements in the sensor field R are governed by a two-dimensional Brownian motion (BM) process with drift – a commonly-employed assumption in the modeling of mobile sensors (Kesidis, Konstantopoulos, & Phoha, 2003). Before presenting the numerical results, we first provide an overview of BM processes. An excellent introduction to this subject can be found in Ross (Ross, 1996).

A continuous-time, continuous-state stochastic process $\{X(t) : t \ge 0\}$ is a one-dimensional BM process with drift on $\mathbb{R}$ if

$$X(t) = \mu t + \sigma B(t),$$

where μ ($\mu \in \mathbb{R}$) is the drift parameter, σ ($\sigma \ge 0$) is the diffusion coefficient, $B(t)$ denotes standard Brownian motion (i.e., $B(t) \sim N(0,t)$ for each $t \ge 0$) and $X(0) = 0$ with probability 1 (w.p. 1). It is well known that $\{X(t) : t \ge 0\}$ possesses stationary and independent increments, and for $0 \le s < t$, the increment $X(t) - X(s)$ is normally distributed with mean $\mu(t-s)$ and variance $\sigma^2(t-s)$. The parameter μ governs the cumulative drift of the process over time, whereas σ scales the random error term $B(t)$. This pattern of mobility combines a linear behavior governed by the drift of the process, as well as randomness governed by Brownian motion (Dousse, Tavoularis, & Thiran, 2006). A bivariate stochastic process $\{(B_x(t), B_y(t)) : t \ge 0\}$ is a two-dimensional BM process if $\{(B_x(t)\}$ and $\{(B_y(t)\}$ are independent and each component is a one-dimensional BM process on $\mathbb{R}$. For our illustrations, we consider mobile sensors moving in the two-dimensional sensor field R according to a two-dimensional BM process with drift. For any time $t \geq 0$, denote the coordinates of sensor j by $(X_j(t), Y_j(t))$, where

$$X_j(t) = \mu t + \sigma B_x(t) \quad \text{and} \quad Y_j(t) = \mu t + \sigma B_y(t),$$

and $\{B_x(t) : t \geq 0\}$ and $\{B_y(t) : t \geq 0\}$ are independent standard BM processes. To compute the quantities q_{jt} of Proposition 1, it is necessary to compute (or estimate) $\mathbb{P}(Z_{ijt} = 1)$, the probability that sensor j is within range of cluster head i at time t. To this end, we simulate the evolution of $\{(X_j(t), Y_j(t)) : t \geq 0\}$ a total of D ($D \in \mathbb{N}$) times and estimate $\mathbb{P}(Z_{ijt} = 1)$ by the proportion of times that node j is within range of cluster head i at time t.

The simulation procedure is described as follows. Initially, we simulate $\xi_1, \ldots, \xi_k$ where $\xi_m \sim N(0,1)$, $m = 1, \ldots, k$. Next, for $t_0 = 0 < t_1 \leq t_2 \leq \cdots \leq t_k$, by exploiting the stationary and independent increments property of BM, sequentially generate $\{X_j(t_m) : m = 1, \ldots, k\}$ as follows for each replication:

$$\begin{aligned}
X_j(t_1) &= \mu t_1 + \xi_1 \sigma \sqrt{t_1} \\
X_j(t_2) &= X_j(t_1) + \mu(t_2 - t_1) + \xi_2 \sigma \sqrt{t_2 - t_1} \\
&= \mu t_1 + \xi_1 \sigma \sqrt{t_1} + \mu(t_2 - t_1) + \xi_2 \sigma \sqrt{t_2 - t_1} \\
&\vdots \\
X_j(t_k) &= \sum_{m=1}^{k} \mu(t_m - t_{m-1}) + \sigma \sum_{m=1}^{k} \xi_m \sqrt{(t_m - t_{m-1})}.
\end{aligned}$$

Similarly, the sequence $\{Y_j(t_m)\colon m = 1, \ldots, k\}$ can be generated for each of the D replications.

For replication n of the simulation, $n = 1, \ldots, D$, let $(X_j^n(t), Y_j^n(t))$ be the position of sensor j at time t, and $(x_i^n(t), y_i^n(t))$ is the position of cluster head i at time t. The Euclidean distance between sensor j and cluster head i at time t in the nth replication is

$$\rho_n^t(i,j) = \sqrt{(X_j^n(t) - x_i^n(t))^2 + (Y_j^n(t) - y_i^n(t))^2}.$$

Define the indicator function

$$\mathbb{1}_{\{\rho_n^t(i,j) \leq r\}} = \begin{cases} 1, & \rho_n^t(i,j) \leq r, \\ 0, & \rho_n^t(i,j) > r, \end{cases}$$

where r is the transmission range of the sensors. Finally, the probability $\mathbb{P}(Z_{ijt} = 1)$ is estimated by

$$\mathbb{P}(Z_{ijt} = 1) \approx D^{-1} \sum_{n=1}^{D} \mathbb{1}_{\{\rho_n^t(i,j) \leq r\}}.$$

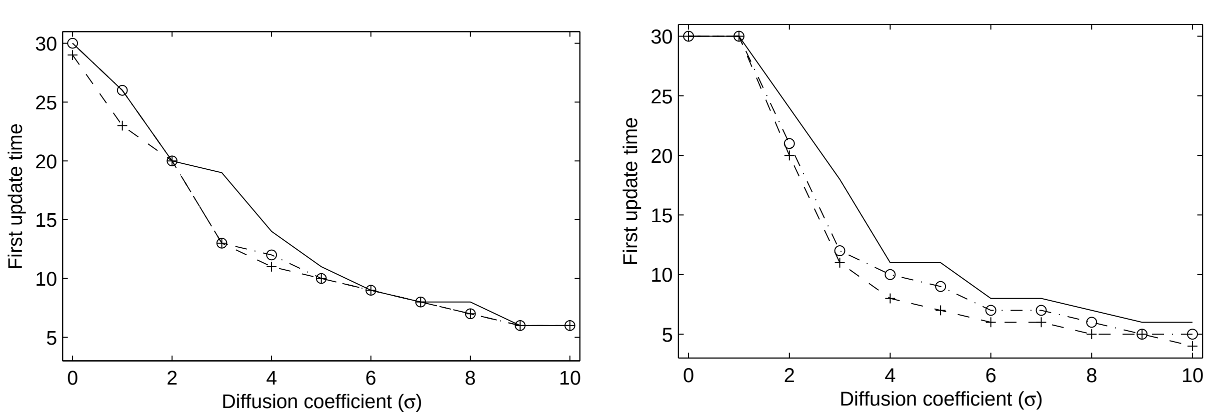
The strong law of large numbers (SLLN) ensures that $D^{-1} \sum_{n=1}^{D} \mathbb{1}_{\{\rho_n^t(i,j) \leq r\}} \to \mathbb{P}(Z_{ijt} = 1)$ w.p. 1 as $D \to \infty$. Empirical testing shows that $D \geq 40$ is sufficient for convergence in our experiments.

Two instances of problem $\mathbf{P}_4$ were solved to optimality to illustrate the usefulness of updating sensor location information over time. Specifically, we present a small problem instance ($N = 25$) and a large problem instance ($N = 200$). The parameter values for each case are summarized in Table 1.

Table 1: Parameter values for small and large problem instances.

Parameter Description	Small Instance	Large Instance
Planning horizon ($\mathcal{T}$)	$\{1, 2, \ldots, 30\}$	$\{1, 2, \ldots, 30\}$
Number of sensors (N)	25	200
Number of candidate cluster head locations (K)	25	196
Maximum number of cluster heads (n)	5	10
Cluster head relocation cost (C)	5	5
Sensor location updating cost (C')	100	100
Per period demand of sensor j (d_j)	$U(10, 20)$	$U(10, 20)$
Probability of link failure (p)	$\{0.1, 0.2, \ldots, 0.9\}$	$\{0.1, 0.2, \ldots, 0.9\}$
Transmission range (r)	13	13
Drift parameter (μ)	1	1
Diffusion coefficient (σ)	$\{0, 1, \ldots, 10\}$	$\{0, 1, \ldots, 10\}$

We plotted the optimal *first* time to update information on the sensor locations as a function of the diffusion coefficient (σ) in Figure 2. The figure is intuitive in that, as the diffusion coefficient σ increases, the variation in the sensor movements increases, thereby reducing the predictability of the sensors' locations. The updating *frequency* is monotone increasing in the diffusion parameter for both the small and large problem instances. Additionally, we note that the optimal first update time decreases as the probability of link failure (p) increases. This is because the gradual decrease in the expected data coverage becomes more pronounced over time. Therefore, information on node locations needs to be updated more frequently to ensure that the rate of total data coverage minus the costs is maximized.



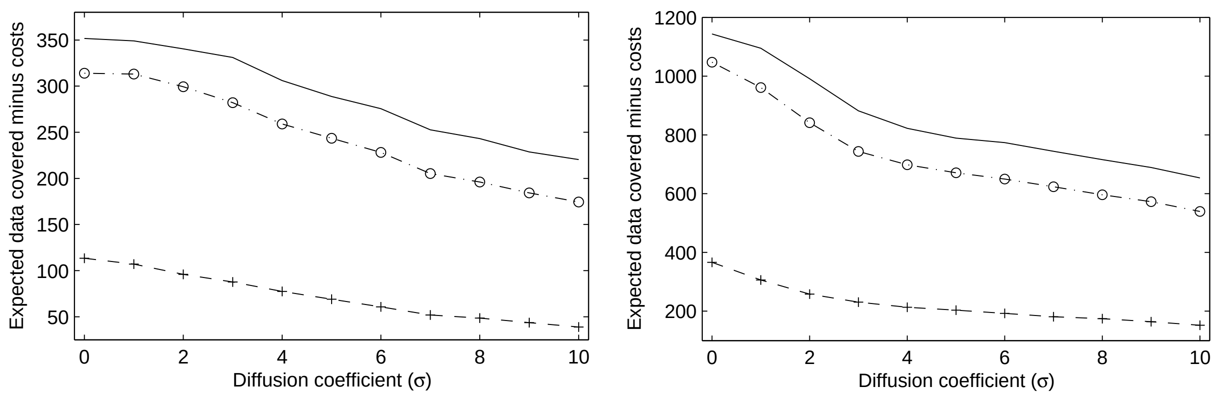
(a) Small problem instance.

(b) Large problem instance.

Figure 2: Optimal first update time when: $p = 0.1$ (–); $p = 0.5$ (·– o –·); $p = 0.9$ (– + –).

Similar conclusions can be drawn from the summarized results of Tables 2 and 3. Interestingly, the objective value is not strictly monotone decreasing in the diffusion coefficient σ for a fixed probability of link failure p, however it is generally decreasing. It is noted that, as the links become more failure-prone (i.e., as p increases), the objective value drops significantly as it is more challenging to cover the demand.

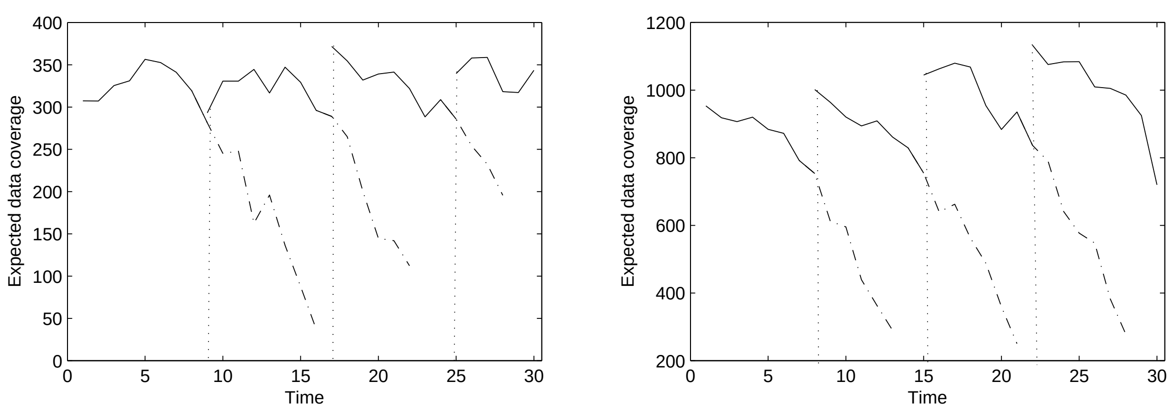
Figure 3 depicts the effects of sensor movement variability and link failure probability on the objective function value. First, we note that as the diffusion coefficient increases, the objective value decreases because, as time progresses, the likelihood that a cluster head is within range of a particular sensor is decreasing. Second, we observe a very substantial decrease in the objective value as the likelihood of link failure (p) increases. These results indicate that link reliability has a substantial impact on data coverage, even when $\sigma = 0$.



(a) Small problem instance. (b) Large problem instance.

Figure 3: Objective function values when: $p = 0.1$ (–); $p = 0.5$ (·– o –·); $p = 0.9$ (– + –).

Figure 4 shows the effect of information updates on the expected coverage. For example, in the small problem instance, exact sensor locations are known at each time $t \in \{0, 9, 17, 25\}$. After these updating times, the expected coverage tends to decrease due to the randomness in the sensor locations. As sensor locations are updated, additional cluster heads are relocated because knowledge of the exact sensor locations leads to opportunities for improving the demand coverage.



(a) Small problem instance. (b) Large problem instance.

Figure 4: Data coverage with location information updates ($\sigma = 6.0$, $p = 0.1$).

The behavior of the first update times as a function of the updating cost (C') and cluster head relocation cost (C) is depicted in Figure 5 for the large problem instance. It is observed that, in the cases of $C = 1$ and $C = 9$, the first update time is insensitive to the cost C'. When the cost of relocating cluster heads is relatively cheap ($C = 1$), the first update time is higher, as the objective is dominated by the updating costs. Similarly, when the cost of relocating cluster heads is relatively high ($C = 9$), it is favorable to gain certainty about the sensor locations earlier in the planning horizon before committing to costly cluster head relocations. However, when $C = 3$, the first update time starts at the lower value 11 and then increases to 13 for $C' \geq 100$. It is conjectured that this jump occurs because, when location updates are expensive, less frequent updating tends to reduce the total costs because the time-averaged, expected updating costs are decreasing in the first update time. The lower graph of Figure 5 illustrates that coverage is monotone decreasing in both relocation and updating costs.

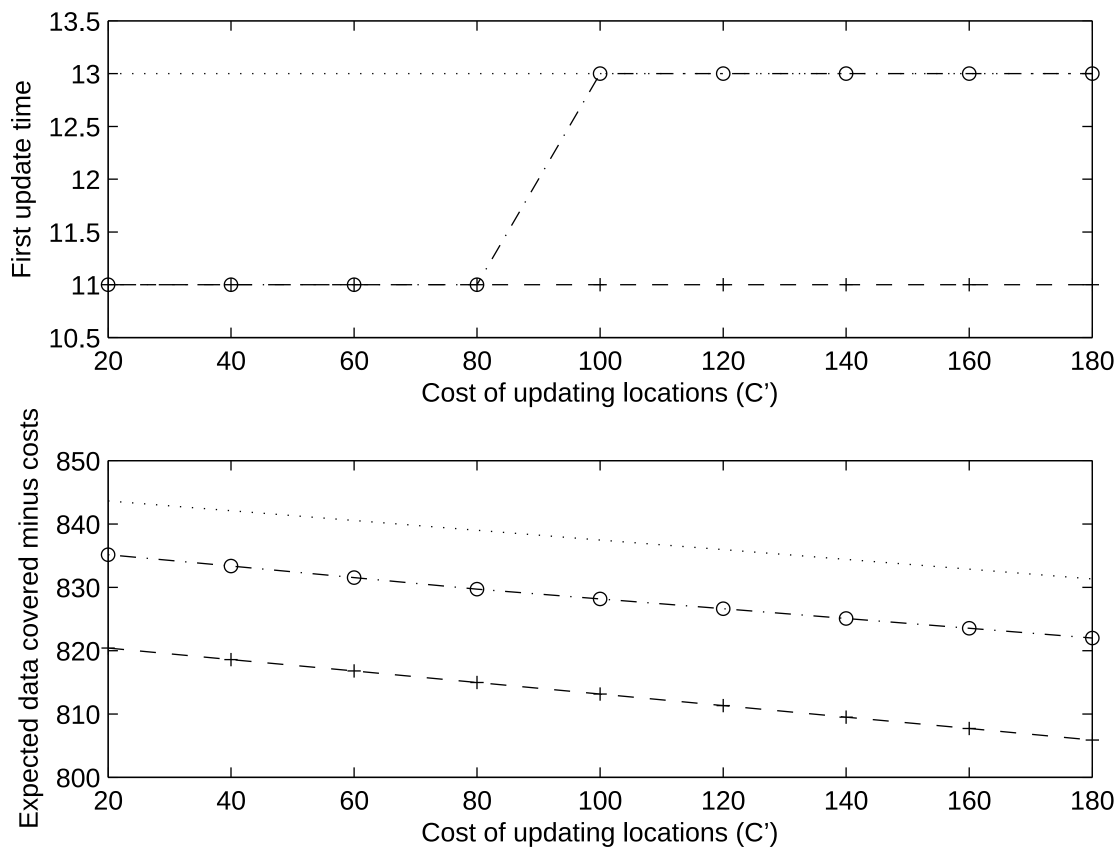


Figure 5: Effect of information update costs on the update time and the objective function value (large problem instance) when $C = 1$ ($\cdots$), $C = 3$ ($\cdot\!-$ o $-\!\cdot$), and $C = 9$ ($-$ + $-$) for $\sigma = 6.0$ and $p = 0.1$.

Table 2: Summary of results for small problem instance.

p	σ	Solution Time (sec)	Objective Value	Optimal Update Time(s)
0.1	0	50.86	351.74	30
0.1	1	64.95	349.01	26
0.1	2	83.37	340.49	20
0.1	3	47.16	338.08	19
0.1	4	29.80	281.46	14
0.1	5	21.92	311.20	11, 21
0.1	6	13.98	294.95	9, 16, 24
0.1	7	13.90	284.31	8, 15, 22
0.1	8	8.50	272.99	8, 15, 22
0.1	9	6.96	253.70	6, 11, 17, 23
0.1	10	4.52	239.89	6, 11, 17, 22
0.5	0	53.40	314.00	30
0.5	1	67.10	313.03	26
0.5	2	83.51	299.43	20
0.5	3	46.94	252.84	13
0.5	4	27.60	267.61	12, 24
0.5	5	18.44	255.31	10, 20
0.5	6	13.10	238.94	9, 16, 24
0.5	7	13.46	225.13	8, 15, 22
0.5	8	8.22	212.38	7, 14, 20
0.5	9	6.82	196.40	6, 11, 17, 23
0.5	10	4.60	182.45	6, 11, 17, 22
0.9	0	61.68	113.45	30
0.9	1	69.94	102.47	23
0.9	2	68.94	95.22	20
0.9	3	46.27	89.64	13
0.9	4	27.41	77.59	11, 23
0.9	5	19.55	69.93	10, 20
0.9	6	13.20	62.64	9, 16, 24
0.9	7	12.36	56.49	8, 15, 22
0.9	8	8.02	50.78	7, 13, 19
0.9	9	6.44	45.03	6, 11, 17, 23
0.9	10	4.46	39.87	5, 9, 14, 19, 23

Table 3: Summary of results for large problem instance.

p	σ	Solution Time (sec)	Objective Value	Optimal Update Time(s)
0.1	0	1371.20	1143.36	30
0.1	1	1372.45	1094.84	30
0.1	2	1317.68	1004.25	24
0.1	3	1257.17	956.47	18
0.1	4	972.46	914.89	11, 22
0.1	5	651.44	861.12	11, 19
0.1	6	427.51	824.48	8, 15, 23
0.1	7	299.49	790.13	8, 14, 19
0.1	8	258.32	755.86	7, 13, 18, 24
0.1	9	253.52	718.91	6, 11, 16, 21
0.1	10	158.12	677.61	6, 10, 15, 19, 24
0.5	0	1364.40	1047.35	30
0.5	1	1370.05	961.05	30
0.5	2	1284.22	875.53	21
0.5	3	1202.54	716.01	12
0.5	4	960.64	745.77	10, 17
0.5	5	637.28	717.89	9, 15, 21
0.5	6	420.40	672.46	7, 13, 18
0.5	7	294.16	646.36	7, 12, 17, 22
0.5	8	252.79	613.68	6, 11, 16, 21
0.5	9	252.10	571.51	5, 9, 13, 17, 21
0.5	10	155.28	543.28	5, 9, 13, 17, 21
0.9	0	1414.03	366.32	30
0.9	1	1358.54	306.39	30
0.9	2	1275.22	271.14	20
0.9	3	1201.55	244.98	11, 21
0.9	4	948.77	225.32	8, 15, 21
0.9	5	625.76	205.33	7, 13, 18, 25
0.9	6	416.81	193.89	6, 12, 16, 21
0.9	7	291.43	184.32	6, 11, 16, 20, 25
0.9	8	250.37	170.79	5, 9, 14, 18, 23
0.9	9	249.67	162.24	5, 9, 14, 18, 23
0.9	10	155.19	150.82	4, 8, 11, 14, 18, 22, 26

CONCLUDING REMARKS

In this paper, we have examined the problem of cluster head location and relocation in mobile wireless sensor networks while focusing on relocation and information updating costs. These methods are especially useful in the military theater where sensors are prone to failure due to disruptions caused by adversaries, or when sensors move out of the communication range of their base stations. The numerical results revealed the effects of certain network parameters, in particular, variability in the movement patterns of sensor nodes and the costs of relocating cluster heads. The proposed optimization models can be solved using commercial solvers; however, heuristic algorithms may be needed to solve these problems for large-scale networks. For example, there may be hundreds, or even thousands, of mobile sensors operating in the sensor field. For such scenarios, it may be preferable to employ simple heuristic algorithms to quickly obtain approximate solutions at each of the updating times.

Future developments will need to focus on optimization models capable of handling more general sensor mobility patterns (we assumed two-dimensional Brownian motion with drift). Furthermore, application-specific algorithms may be required to improve the WSN lifetime and coverage when sensors move according to a specification of the network. Other than coverage and relocation costs, cluster head location decisions may also need to account for uniform energy consumption and quality-of-service considerations. Finally, an important extension of the model is to examine the optimal location and relocation of cluster heads when data aggregation is performed by a mobile agent.

ACKNOWLEDGEMENTS

The authors are grateful to two anonymous referees, whose suggestions have improved the content and presentation of this work. G. Degirmenci and J.P. Kharoufeh were supported, in part, by a grant from the U.S. National Science Foundation (CNS-0831707). O.A. Prokopyev was partially supported by the U.S. Air Force Office of Scientific Research (FA9550-11-1-0037) and the U.S. Air Force Summer Faculty Fellowship.

REFERENCES

Abbasi, A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, *30*(14–15), 2826–2841.

Aioffi, W., Valle, C., Mateus, G., & Cunha, A. (2011). Balancing message delivery latency and network lifetime through an integrated model for clustering and routing in wireless sensor networks. *Computer Networks*, *55*(13), 2803–2820.

Ci, S., Guizani, M., & Sharif, H. (2007). Adaptive clustering in wireless sensor networks by mining sensor energy data. *Computer Communications*, *30*(14–15), 2968–2975.

Dimokas, N., Katsaros, D., & Manolopoulos, Y. (2010). Energy-efficient distributed clustering in wireless sensor networks. *Journal of Parallel and Distributed Computing*, *70*(4), 371–383.

Ding, P., Holliday, J., & Celik, A. (2005). Distributed energy-efficient hierarchical clustering for wireless sensor networks. In V. Prasanna, S. Iyengar, P. Spirakis, & M. Welsh (Eds.), *Distributed Computing in Sensor Systems* (Vol. 3560, p. 322-339). Berlin, Germany : Springer-Verlag.

Dousse, O., Tavoularis, C., & Thiran, P. (2006). Delay of intrusion detection in wireless sensor networks. In *Proceedings of the 7th ACM international Symposium on Mobile Ad Hoc Networking and Computing* (pp. 155–165).

Furuta, T., Sasaki, M., Ishizaki, F., Suzuki, A., & Miyazawa, H. (2009). A new clustering model of wireless sensor networks using facility location theory. *Journal of the Operations Research Society of Japan*, *52*(4), 366-376.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, *22*(4), 455–460.

Gupta, I., Riordan, D., & Sampalli, S. (2005). Cluster-head election using fuzzy logic for wireless sensor networks. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference* (pp. 255–260).

Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)* (pp. 1–10).

Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transaction on Wireless Communications*, *1*(4), 660–670.

ILOG CPLEX. (2015). `http://www.ilog.com/products/cplex`. (Accessed: 2015-02-16.)

Islam, A., Hyder, C., Kabir, H., & Naznin, M. (2010). Stable sensor network (SSN): A dynamic clustering technique for maximizing stability in wireless sensor networks. *Wireless Sensor Network*, *2*(7), 538–554.

Kesidis, G., Konstantopoulos, T., & Phoha, S. (2003). Surveillance coverage of sensor networks under a random mobility strategy. In *Proceedings of IEEE Sensors, 2003* (Vol. 2, pp. 961–965).

Khan, A., Madani, S., Hayat, K., & Khan, S. (2012). Clustering-based power-controlled routing for mobile wireless sensor networks. *International Journal of Communication Systems*, *25*(4), 529–542.

Koucheryavy, A., & Salim, A. (2010). Prediction-based clustering algorithm for mobile wireless sensor networks. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT)* (Vol. 2, pp. 1209–1215).

Krivitski, D., Schuster, A., & Wolff, R. (2005). A local facility location algorithm for sensor networks. In V. Prasanna, S. Iyengar, P. Spirakis, & M. Welsh (Eds.), *Distributed Computing in Sensor Systems* (Vol. 3560, pp. 368–375). Berlin, Germany : Springer-Verlag.

Kumar, V., Jain, S., & Tiwari, S. (2011). Energy efficient clustering algorithms in wireless sensor networks: A survey. *International Journal of Computer Science Issues*, *8*(5), 259–268.

Liu, X., & Shi, J. (2012). Clustering routing algorithms in wireless sensor networks: An overview. *KSII Transactions on Internet and Information Systems*, *6*(7), 1735–1755.

Manisekaran, S., Venkatesan, R., & Deivanai, G. (2011). Mobile adaptive distributed clustering algorithm for wireless sensor networks. *International Journal of Computer Applications*, *20*(7), 12–19.

Patel, D. J., Batta, R., & Nagi, R. (2005). Clustering sensors in wireless ad hoc networks operating in a threat environment. *Operations Research*, *53*(3), 432–442.

Ross, S. (1996). *Stochastic Processes, Second Edition.* New York, NY : John Wiley & Sons, Inc.

Sevgi, C., & Kocyigit, A. (2009). An optimal network dimensioning and initial energy assignment minimizing the monetary cost of a heterogeneous WSN. In *Proceedings of the 6th International Symposium on Wireless Communication Systems* (pp. 517–521).

Shanbehzadeh, J., Mehrjoo, S., & Sarrafzadeh, A. (2011). An intelligent energy efficient clustering in wireless sensor networks. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1, pp. 2–6).

Slama, I., Ghedira, M., Jouaber, B., & Afifi, H. (2007). Cluster based wireless sensor networks' optimization under energy constraints. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information* (pp. 745–750).

Wang, G., Wang, T., Jia, W., Guo, M., Chen, H., & Guizani, M. (2007). Local update-based routing protocol in wireless sensor networks with mobile sinks. In *Proceedings of the IEEE International Conference on Communications* (pp. 3094–3099).

Xia, D., & Vlajic, N. (2007). Near-optimal node clustering in wireless sensor networks for environment monitoring. In *Proceedings of the 21st International Conference on Advanced Networking and Applications* (pp. 43–50).

Younis, O., & Fahmy, S. (2004). HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, *3*(4), 366–379.

Younis, O., Krunz, M., & Ramasubramanian, S. (2006). Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, *20*(3), 20–25.

Youssef, M., Youssef, A., & Younis, M. (2009). Overlapping multihop clustering for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, *20*(12), 1844–1856.

Yu, M., Leung, K., & Malvankar, A. (2007). A dynamic clustering and energy efficient routing technique for sensor networks. *IEEE Transactions on Wireless Communications*, *6*(8), 3069-3079.

Zhong, T., Wang, S., Xu, S., Yu, H., & Xu, D. (2007). Time delay based clustering in wireless sensor networks. In *Proceedings of the Wireless Communication and Networking Conference* (pp. 3959–3963).