

Quality of Monitoring of Stochastic Events by Periodic and Proportional-Share Scheduling of Sensor Coverage

DAVID K. Y. YAU, NUNG KWAN YIP, and CHRIS Y. T. MA

Purdue University

and

NAGESWARA S. V. RAO and MALLIKARJUN SHANKAR

Oak Ridge National Lab

We analyze the quality of monitoring (QoM) of stochastic events by a periodic sensor which monitors a point of interest (PoI) for q time every p time. We show how the amount of information captured at a PoI is affected by the proportion q/p , the time interval p over which the proportion is achieved, the event type in terms of its stochastic arrival dynamics and staying times and the utility function. The periodic PoI sensor schedule happens in two broad contexts. In the case of static sensors, a sensor monitoring a PoI may be periodically turned off to conserve energy, thereby extending the lifetime of the monitoring until the sensor can be recharged or replaced. In the case of mobile sensors, a sensor may move between the PoIs in a repeating visit schedule. In this case, the PoIs may vary in importance, and the scheduling objective is to distribute the sensor's coverage time in proportion to the importance levels of the PoIs. Based on our QoM analysis, we optimize a class of periodic mobile coverage schedules that can achieve such proportional sharing while maximizing the QoM of the total system.

A preliminary version of this article appeared in *Proceedings of ACM CoNEXT*, ACM, New York, 2008.

The research was supported in part by the U.S. Department of Energy under SensorNet grant no. AC05-00OR22725 and Mathematics of Complex, Distributed, Interconnected Systems program, Office of Advanced Computing Research, in part by the U.S. National Science Foundation under grant numbers DMS-0707926 and CNS-0964086, and in part by the Laboratory of Directed Research and Development program at Oak Ridge National Laboratory (ORNL). Work was performed at Purdue University and ORNL managed by UT-Battelle, LLC.

Authors' address: D. K. Y. Yau, N. K. Yip, and C. Y. T. Ma, 305 North University Street, West Lafayette, IN 47907; email: {yau,yipn,ma18}@cs.purdue.edu; N. S. V. Rao and M. Shankar, Oak Ridge National Laboratory, 1 Bethel Valley Road, Bldg. 5600, Oak Ridge, TN 37831; email: {raons, shankarm}@ornl.gov.

© 2010 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the [U.S.] Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1550-4859/2010/08-ART18 \$10.00
DOI 10.1145/1824766.1824774 <http://doi.acm.org/10.1145/1824766.1824774>

Categories and Subject Descriptors: C.4 [**Computer System Organization**]: Performance of Systems; G.1.6 [**Numerical Analysis**]: Optimization

General Terms: Performance, Theory

Additional Key Words and Phrases: Sensor network, mobile coverage, periodic scheduling, proportional sharing

ACM Reference Format:

Yau, D. K. Y., Yip, N. K., Ma, C. Y. T., Rao, N. S. V., and Shankar, M. 2010. Quality of monitoring of stochastic events by periodic and proportional-share scheduling of sensor coverage. *ACM Trans. Sensor Netw.* 7, 2, Article 18 (August 2010), 49 pages.

DOI = 10.1145/1824766.1824774 <http://doi.acm.org/10.1145/1824766.1824774>

1. INTRODUCTION

There is considerable interest in using sensors to protect populated areas against physical hazards, such as chemical, biological, nuclear, radiational, and explosive (CBNRE) leaks/attacks. Real-world sensors have limited ranges of tens to hundreds of feet. If the area to be protected is large, it may be difficult to install and deploy a sufficient number of sensors to cover the entire area. This leads to strong interest in the use of mobile sensors to expand the area of coverage, so that one sensor can cover multiple *points of interest* (PoIs) where interesting events may dynamically appear and disappear.

Note that there are many real-world examples of mobility in monitoring tasks. In traditional public safety work, policemen or security guards are on patrol schedules around town or inside facilities to detect crimes. In national security, reconnaissance planes routinely fly over mission areas to collect surveillance images, since the installation of (static) video cameras in the mission areas may be out of the question (e.g., they are foreign or enemy territories). In the case of sensor networks, certain sensors are expensive and complete area coverage by static sensors would have prohibitive costs. For example, in the Memphis Port deployment against water pollution/poisoning [Lee and Kulesz 2006], the engineers emphasize in the project report that with the high procurement, installation, and management costs of the Smith APD2000 chemical sensors, it was not possible to have complete area coverage. They then made the difficult decision to (statically) place the sensors where the impact on people protection is highest.

There are also situations in which, independent of costs, mobility is simply required for the sensor network. For example, when (static) sensors are placed at PoIs where long-range data communication is difficult (e.g., underwater [Bisnik et al. 2006], where the high wireless signal attenuation in water makes it infeasible to transmit sensor data over long distances, or in an underground system of ducts where complex pathways connect the PoIs so that the placement of communication nodes to reliably get data out from underground is extremely hard), a mobile node is necessary to move between the PoIs to collect the sensor data and carry them to a data center for analysis. In this case, data may be buffered at a sensor before they are read, but the buffer capacity is limited so that unread data may be replaced by newer data and lost. Hence, the data available for reading are similar to stochastic events that stay for a

time duration after which they will disappear.¹ The use of a mobile node for data collection also has the advantage of reducing the sending energy of the sensors [Chakraborty et al. 2006].

At the same time, it is recognized that different parts of the protected area may vary in terms of their importance. For example, as in the Memphis Port deployment, some parts are densely populated while other parts are sparsely populated, so that an undetected hazard in the former will result in more harm than in the latter. In such a scenario, simple area of coverage is no longer sufficient. An arguably more suitable goal is to allocate sensing resources to the different parts in proportion to their importance levels. Note that in the case of *static* sensors, their placement to best protect people has been considered in the Memphis Port deployment [Lee and Kulesz 2006]. There, because the Smith APD2000 chemical sensors used to detect toxic chemical leaks are expensive, they cannot cover the whole area. A search method is used to best place the next sensor to maximize the marginal increase in the number of people protected. Proportional sharing of resources is not a new concept. The notion has been employed in the scheduling of CPU time, network bandwidth, buffers, disk space, etc [Jeffay et al. 1998; Parekh and Gallager 1993], where the performance impact on the rates of computation and communication has been well studied. In CPU scheduling, for example, a scheduler may give one task twice the CPU share as another task. In this case, the performance impact is more or less clear: the first task gets twice as much computation done as the other task over the same real-time interval, if both tasks run the same application. In the case of sensor coverage, however, proportional sharing must be evaluated in terms of its impact on the *quality of monitoring* (QoM), which can be expressed as the number of interesting events captured, or the total amount of information captured about these events. The problem is not well understood.

In this article, we target the problem of information capture about stochastic events (e.g., a chemical leak) that dynamically appear and disappear at a given set of locations called *points of interest* (PoI). The events are detected by a mobile sensor (e.g., a chemical detector carried by a robot) which allocates its coverage time among the PoIs in proportion to their importance levels. In our problem, we argue that the QoM of proportional-share sensor coverage may not have a simple interpretation that γ times the resource allocation to a PoI will result in γ times better performance for the PoI. Rather, the achieved QoM is an interesting function of several important system parameters, including the time scale of the proportional sharing, the event dynamics, and the type of events. Our contributions are two fold, summarized as follows.

First, we provide extensive analysis to answer the following questions as a function of the event dynamics and type of events: (1) What is the QoM of a sensor that covers a PoI for q time every p time? Does a higher proportional

¹The sensing range in Section 2 will then correspond to the communication range between the sensor and the mobile node. The event utility function there might account for the time needed for different useful fractions of the total data (about a physical world event) to be uploaded from the sensor to the mobile node.

share of q/p imply a linearly higher QoM? (2) For the same q/p , what is the impact of the period p that controls the fairness granularity of the proportional sharing? Under what situations is finer/coarser time-scale sharing preferred over the other? (3) What is the scaling law of mobile coverage, that is, when a mobile sensor is allocated among k out of n PoIs, how is the average QoM over all the PoIs affected as k increases? Can mobility fundamentally improve the sensing by increasing the achievable QoM?

Second, based on the QoM analysis, we will analyze the performance of a class of periodic coverage algorithms considering the travel time overhead between PoIs. We first optimize a *linear periodic* sensor schedule for maximum total QoM that achieves given proportional shares of the coverage time. We then discuss the optimization of general periodic schedules. We present a *simulated annealing* algorithm for finding a solution close to the global optimal with high probability and within a practical time budget.

We mention in passing that independent of mobility, the analysis of periodic sensor schedules has obvious applications in energy-efficient sensing. In this case, an energy-constrained *static* sensor may be deployed at each PoI, and there is a need to periodically turn off the sensor to conserve energy, so that the sensor will last long enough until it can be recharged or replaced. Our analysis in Section 4 gives directly the QoM of such a periodic sensor. In particular, our results show that for events that stay, the QoM of a sensor working for q/p of the time may capture a fraction of information much higher than q/p . Hence, such periodic scheduling of the sensors can be quite productive. Our results also show where it is useful (and where it is not) to have finer granularity of the periodic scheduling, in terms of a smaller p , to achieve a higher QoM. In this case, however, it is clear that the benefits of extremely fine grained periodic scheduling may be limited/offset by the latency and energy costs of turning on/off a sensor. The development of the full details such as the energy models is out of the scope of the present article. Also, for simplicity of exposition, we will develop our problem from the point of view of mobile coverage only in the rest of this article.

2. PROBLEM STATEMENT

We assume that events appear and disappear at given points of interest (PoIs) and are to be monitored by a sensor whose sensing range is R and whose sensing region is a circle of radius R . Although such a “perfect disk” sensing model is widely applied, it can be a simplifying assumption as the coverage regions of real-world signals have been found to be nonisotropic [He et al. 2003]. However, while it is possible to obtain more accurate numerical solutions by considering more elaborate sensing models (e.g., accounting for the exact geometry of the sensing region or specifying the sensing range as a random variable of some probability distribution), the simple model will allow us to obtain essential results about how event monitoring is impacted by the event types and dynamics, without being detracted by the more involved mathematics. The PoIs are located on a 2D plane. A pair of PoIs, say i and j , are connected by a road, given by E_{ij} , of distance d_{ij} . If there is no road that directly connects i and j ,

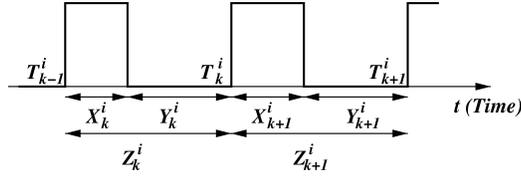


Fig. 1. Event dynamics at PoI i , with the event staying and absent times X_k^i and Y_k^i for $k = 1, 2, \dots$

$d_{ij} = \infty$. Otherwise, the sensor traveling at speed v from i to j takes time d_{ij}/v to complete the trip.

The next set of assumptions concerns the event dynamics. The events appear at PoI i one after another. After appearing, each event stays for a duration of time, which we call the *event staying time*, and then disappears. The next event appears after another duration of time, which we call the *event absent time*. Here and in the following, we will use superscript and subscript to denote the PoI and event indices. We denote the sequential staying and absent times by $\{X_k^i\}_{k \geq 1}$ and $\{Y_k^i\}_{k \geq 1}$. The *event inter-arrival time* is then denoted by $Z_k^i = X_k^i + Y_k^i$. We assume that (for each i) the vectors $\{(X_k^i, Y_k^i)\}_{k \geq 1}$ are i.i.d. random variables drawn from a common distribution (X^i, Y^i) , even though for each k , the X_k^i and Y_k^i may be dependent. Lastly, the commonly known *event arrival times* can be recovered by the formula: $T_0^i = 0$, $T_k^i = T_{k-1}^i + Z_k^i$ for $k, i \geq 1$ even though the T_k^i 's will not be used in the analysis. These variables are illustrated in Figure 1.

An important assumption behind the analysis of this article is that the event dynamics at different PoIs are independent. This can be justified in two situations: (i) there are indeed no correlations between the PoIs because they are physically isolated or are far apart relative to the spatial extents of the events (e.g., monitoring two storage tanks of chemicals in two separate rooms for leakage); and (ii) regardless of the presence or absence of correlations between the PoIs, the information acquired at different PoIs might not be aggregated but is accounted for on a *per-PoI basis* (e.g., a manager at an operation facility X in a factory is satisfied by her knowledge about X, regardless of whether similar or the same information is also acquired by a manager at another facility Y). Clearly, if information can be aggregated across PoIs to gain further global information about a target (e.g., movement of the target in a global surveillance area), this assumption will have to be relaxed, but such aggregation is beyond the scope of the current article.

We further classify the events as follows. When the staying time drawn from X^i is an infinitesimally small amount of time ϵ , the corresponding events are like “blips,” that is, they do not stay but disappear instantaneously after arrival. Another type of events is that which stays, that is, there is an $0 < \epsilon \ll 1$ such that $P(X \geq \epsilon) = 1$.

An event at a PoI is captured by the sensor provided that the PoI is within range of the sensor during the event’s lifetime. We assume that events are *identifiable*, that is, when the sensor sees an event at a PoI, leaves the PoI, but comes back later to see the same event, it will know that it is the same

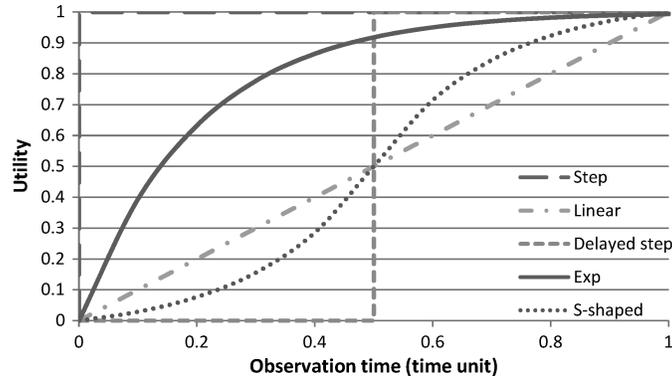


Fig. 2. The utility functions.

event. It is important to note that many investigative actions in the real-world assume the identifiability of events naturally, either by reasonable domain knowledge (e.g., if the spread of an oil spill in the ocean is monitored by periodic measurements of the oil concentrations at specific locations, it is reasonable to assume that the same oil spill is being monitored unless the oil spill is known to have been cleaned up) or because the target possesses easily identifiable attributes (e.g., when a doctor orders diagnostic tests for a patient, clearly the identity of the patient is assumed known, and the doctor accumulates knowledge about possible diseases by a series of non-contiguous tests). On the other hand, if events are not identifiable, then it is not possible to accumulate knowledge across noncontiguous measurements and the information learned for an event will be in principle the *maximum* obtained for that event among all the individual measurements. However, in practice, the set of measurements that should be used in taking the maximum is not clear without event identities. The monitoring of such nonidentifiable events is beyond the scope of this article. We assume that as the sensor observes an event, the information it accumulates about the event is nondecreasing as the observation time increases. We quantify the sensing quality as a utility function that increases monotonically from zero to one as a function of the total observation time. Figure 2 illustrates the following five examples of the utility function considered in this article:

(a) *Step Function*: $U_I(x) = 1$ for $x > 0$. Full information about an event is obtained instantaneously on detection. This function might approximate the detection of events that have easily observable attributes, for example, determining the colors (rather than more detailed properties) of cars passing by a checkpoint using a video feed. It is also widely adopted in the sensor network literature, where events are assumed captured whenever they fall within the sensing range of a sensor, no matter how brief the sensing time is.

(b) *Exponential Function*: $U_E(x) = 1 - e^{-Ax}$. Much of the information about an event is obtained at the beginning but the marginal gain decreases as the observation time gets longer. This function models events about which the

gain in confidence with further measurements follows the law of diminishing marginal returns, such as the detection of radioactive sources [Sundaresan et al. 2007].

(c) *Linear Function*: $U_L(x) = Mx$ for $0 \leq x \leq \frac{1}{M}$ and $U_L(x) = 1$ for $x \geq \frac{1}{M}$. Information obtained increases linearly with the observation time until the full information is achieved. A real-life example might be, in solving a crime case, identifying the people who entered a room within a time window, where there are no known correlations between who might enter or not. In this case, each identification contributes incrementally to the overall understanding, and they are equally important.

(d) *S-Shaped Function*: $U_S(x)$. The initial observation gains little information until a critical observation time is reached, at which point there is a large marginal gain of information in a short time, and afterwards the marginal gain drops sharply as the full information is approached. This function models general learning trends which are combinations of positive and negative learning curves [Cochran 1960]. Once a critical mass of basic knowledge is obtained, the learning of a skill improves quickly with some more training, but once a threshold is reached, perfection of the skill may take a very long time.

(e) *Delayed Step Function*: $U_D(x) = U_I(x - D)$. No information is gained until the total time of observation exceeds a threshold value D , after which the full information is captured instantaneously. A real-life example might be the examination of a product for certification, and certification is given only if a required sequence of tests are all passed. The delayed step function is also an approximation for the S-shaped function but is more amenable to analysis. We view (a) and (e) as extreme cases. All of the above, except (d), are quite amenable to analytical formulations.

When PoI i falls within the range of the sensor, we say that the sensor is *present* at i . Otherwise, the sensor is *absent* from i . Since we are interested in the resource competition between different PoIs, we make the following assumption.

Assumption 1. The PoIs and the roads between them are separated such that (1) no two PoIs fall within the range of the sensor at the same time; (2) for the sensor traveling from PoI i to PoI j on E_{ij} at speed v , i will be within range of the sensor for R/v time before the sensor leaves i , and j will be within range of the sensor for R/v time until the sensor reaches j , and (3) no PoI other than i and j falls within the range of the sensor during the trip on E_{ij} . In general, however, the sensor can vary its speed while traveling on a road.

2.1 Definition of QoM

We now define the quantitative measurement of the QoM at a PoI or for the whole protected area. In the course of a deployment, denote by $e_1^i, \dots, e_{t_i}^i$ the sequence of events appearing at PoI i over the duration $[0, T]$ of the deployment. For the event e_j^i , assume that it is within range of the sensor for a total (but not necessarily contiguous) amount of time t_j^i , where $t_j^i \geq 0$. The sensor will

then gain a certain amount of information, $U_j^i(t_j^i)$, about e_j^i , where $U_j^i(\cdot)$ is the utility function of e_j^i . The total information gained by the sensor at i is defined by $E_i(T) = \sum_{1 \leq j \leq l_i} U_j^i(t_j^i)$, and the average information gained per event at i during the whole deployment period is then $\bar{E}_i(T) = E_i(T)/l_i$. Similarly, the total information gained by the sensor in the whole deployment is $E_*(T) = \sum_{1 \leq i \leq n} E_i(T)$, where n is the number of PoIs in the protected area. The average information gained per event in the whole deployment is then

$$\bar{E}_*(T) = \frac{1}{\sum_{1 \leq i \leq n} l_i} \left(\sum_{1 \leq i \leq n} l_i \bar{E}_i(T) \right).$$

By means of the strong law of large numbers and renewal theory, $\bar{E}_i(T)$ and $\bar{E}_*(T)$ can be shown to converge to a deterministic number as $T \rightarrow \infty$. Hence, we define the QoM of PoI i and the whole covered area as:

$$Q_i = \lim_{T \rightarrow \infty} \bar{E}_i(T), \text{ and } Q_* = \lim_{T \rightarrow \infty} \bar{E}_*(T). \quad (1)$$

Furthermore, they are related by:

$$Q_* = \frac{1}{\mu_*} \sum_{1 \leq i \leq n} \mu_i Q_i, \quad (2)$$

where $\mu_i = \frac{1}{\bar{E}(Z)}$ is the mean event arrival rate at PoI i and $\mu_* = \sum_{1 \leq i \leq n} \mu_i$.

Remark. Note that in defining the QoM, we should in principle divide not only by the number of events l , but also by the maximum possible utility achievable for an event: $\int_0^\infty U(x) f(x) dx$, where $f(x)$ is the pdf of the event staying time distribution. The latter may be less than 1 if the events do not stay infinitely long. However, the difference is only by a proportionality constant, and hence will not affect our comparison results. Unless otherwise stated, we will further assume that all the events at i have the same utility function, and denote this function by $U^i(\cdot)$.

3. RELATED WORK

Quality of monitoring metrics in a sensor network have been proposed, for example, the rate of false positives in a detection problem [He and Hou 2005]. Area coverage in a sensor network has been well studied [Gupta et al. 2003; Meguerdichian et al. 2001; Wang et al. 2003]. Protocols have been proposed to task subsets of sensors in a dense network to provide maximum lifetime area coverage [Zhang and Hou 2005]. Simple area coverage does not consider the varying importance of different subregions. Our work addresses the heterogeneity of sub-regions by proportional-share coverage. Proportional-share resource allocation has been proposed for CPU/OS scheduling [Jeffay et al. 1998; Waldspurger and Weihl 1994], and network scheduling for both bandwidth and queue buffers [Parekh and Gallager 1993]. Mobile coverage has the additional challenge that the sensor schedules can be severely constrained by the adjacencies of and distances between the PoIs.

The importance of the sensing time in accurately assessing various physical phenomena has been well documented [Lapp and Andrews 1948]. The need for

nonnegligible sensing durations to obtain useful information is due to noises in the measurement process and the probabilistic nature of the phenomena under observation. The impact of the sensing time on the information gained is captured by the event utility functions in our problem statement.

Mobility has been discussed in delay-tolerant networks, vehicular networks, and sensor networks. It has been shown to improve the coverage of sensor networks when used either throughout a deployment [Liu et al. 2005] or in a more limited form during the initial phases of deployments only [Howard et al. 2002; Zou and Chakrabarty 2003; Wang et al. 2004; Poduri and Sukhatme 2004]. Passive mobility has been analyzed for its effects on providing communication opportunities [Hull et al. 2006; Zhang et al. 2007], and carry-and-forward network protocols have been proposed [Costa et al. 2006; Dai et al. 2007]. Mobility control has been used to deploy ferries and data mules among a number of data sources, to optimize communication of the source data to the data sink [Shah et al. 2003; Zhao et al. 1994]. In a hybrid mobile/static sensor network, similar data mules are useful for collecting and disseminating data reports from the static sensors to a control center [Wang et al. 2005]. Route optimization of ferries/mules is in general NP hard, and various heuristic algorithms have been designed and shown to be effective.

Various problems of using mobile sensors for area monitoring have been studied. Carlsson et al. [1993] study the problem of finding the shortest watchman route in an area which is a simple polygon. They aim to find a route inside the polygon such that every point inside the polygon can be observed by a point on the route. Carlsson et al. [1991] study a variation of the problem for multiple watchmen in which each watchman picks its own shortest route. Efrat et al. [2000] study the problem of event detection by deploying mobile “guards” to sweep a region under the requirements that the guards follow the same route and that consecutive guards along the chain are mutually visible. Guibas et al. [1999] study the problem of using mobile pursuers to capture a mobile evader by vision. Batalin and Sukhatme [2003] study the use of mobile sensors for area patrol and exploration. The sensors need to patrol all given areas as frequently as possible, but the monitoring time at each point is irrelevant. They propose a distributed algorithm in which static beacons are deployed by the mobile sensors for assistance. These problems are all different from ours in that our PoIs are of different importance and the quality of information obtained in our problem is affected by the temporal dimension of the sensing.

The dynamics of real-world events are frequently modeled as stochastic processes. Poisson arrivals are generally accurate characterizations of a large number of independent event occurrences, whose event inter-arrival times are Exponentially distributed. Real-world network/computing workloads have properties that are found to be long-range dependent and follow the Pareto distribution, for example, the distribution of file sizes in a file system [Gribble et al. 1998], or the distribution of traffic in a computer network [Crovella and Bestavros 1996; Leland et al. 1993]. In a sensor network, the target events may have similar dynamic behaviors. For example, radioactive particles arriving at a Geiger-Müller counter follow a Poisson process [Lapp and Andrews 1948]; a chemical leak at a facility may occur with a probability, and the leak may

persist for a random duration until the chemical has been dispersed; people may arrive at a location and then leave. Our analysis applies to a wide range of event inter-arrival and staying time distributions.

The monitoring of PoIs by mobile sensors has been studied in Cheng et al. [2008], Gupta et al. [2006], and Bisnik et al. [2006]. The concept of sweep coverage is introduced in Cheng et al. [2008] and contrasted with static full coverage and barrier coverage problems in the literature. In sweep coverage, the PoIs need to be visited once every given time interval, and centralized and distributed algorithms are proposed to satisfy the requirement. Gupta et al. [2006] study the problem of finding the minimum number of sensors to estimate the state of processes present in a network with a bounded estimate error covariance. They assume that the measurements made by the sensors are coupled with zero mean white Gaussian noise. The consideration of explicit stochastic events at PoIs in the mobile sensing has been studied in Bisnik et al. [2006]. Our problem in this article is quite different from these earlier papers. First, we consider differential coverage of PoIs by proportional sharing whereas they do not. In particular, we analyze the QoM of periodic sensor schedules, as a function of the proportional share q/p and the period p . Such analysis has applications besides mobile coverage, for example, energy-efficient sensing by periodically turning off a sensor. Second, we consider sensing tasks with the temporal dimension as defined by the event utility function, whereas they either do not consider the events explicitly or focus only on the number of captured events (where an event is captured whenever it falls within the sensing range of a sensor). Third, we define the concepts of linear and general periodic schedules among the PoIs, and design optimal algorithms for both kinds of schedule.

4. SINGLE-POI ANALYSIS OF QOM

This section forms the basis of the analysis of the impact on the QoM by the coverage schedule of a sensor at a given PoI. The schedule specifies the time intervals over which the sensor is present at or absent from the PoI. A given schedule is achieved by how the sensor moves between the PoIs according to some movement algorithm. The problem of the algorithm design and the feasibility of a set of PoI schedules are the subject of Section 5.

We can already illustrate some interesting QoM properties of proportional-share mobile coverage by considering only *periodic* schedules at individual PoIs. Specifically, we assume that the sensor is alternately present and absent at a PoI, say i , for q^i and $p^i - q^i$ time units, respectively. For example, let S_1 be the following the coverage schedule of i :

$$S_1 = \{PAAAPAAA\dots\}$$

for $q^i = 1$ and $p^i = 4$. In the schedule, P denotes one time unit of the sensor's presence and A denotes one time unit of the sensor's absence. Thus, the proportional share equals $q^i/p^i = 25\%$ of the sensor's total coverage time.

Clearly, a given proportional share for i can be achieved in many different ways. For example, $q^i = 2$ and $p^i = 8$ give the following schedule S_2 with the

same 25% share for i :

$$S_2 = \{PPAAAAAAPAAAAA\dots\}.$$

While S_1 and S_2 are equivalent from the proportional-share point of view, they differ in terms of the time scale over which the proportional share is achieved. Specifically, S_1 achieves the 25% share over a time period of 4 time units, whereas S_2 achieves the same share over a period of 8 time units. We say that S_1 has a finer *fairness granularity* than S_2 , and will use p_i to quantify this fairness granularity. Notice that for a fixed proportional share, a smaller p_i implies a proportionately smaller q_i .

The main purpose of this section is to analyze the dependence of the QoM on the utility function and the fairness granularity. In this section, as we will focus on a single PoI, the superscript i will be omitted where there is no confusion.

The problem as formulated in Section 2 fits perfectly well in the realm of renewal theory (see Ross [1996, Chapter 3]). Recall that T_k refers to the k th event arrival time. Then the function $N([0, t]) = \sum_{k=1}^{\infty} 1_{[0, t]}(T_k)$ is the total number of arrivals in the interval $[0, t]$. Its expectation $m(t) = E(N([0, t]))$ is called the *renewal function*. Many important quantities about the renewal process $\{T_k\}_{k \geq 1}$ can be expressed in terms of $m(\cdot)$. In the following, we use $\mu = 1/E(Z_k)$ to denote the event arrival rate. The main results from renewal theory we will use are:

- (1) $\lim_{t \rightarrow \infty} \frac{N([0, t])}{t} = \mu$ a.s.;
- (2) $\lim_{t \rightarrow \infty} \frac{m(t)}{t} = \mu$;
- (3) $\lim_{t \rightarrow \infty} m(t + a) - m(t) = \mu a$, for any $a > 0$.

The last statement is true provided that the distribution of Z is not lattice. It shows that regardless of the distribution of Z , in the long run, the “probability” of an event arriving in an interval dt equals μdt .

The following two types of event staying time distribution will be considered in this article, where $f(x)$ is the pdf of X .

—Exponential Distribution ($\lambda > 0$):

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \text{mean} = \frac{1}{\lambda}.$$

—Pareto Distribution ($\alpha, \beta > 0$):

$$f(x) = \frac{\alpha \beta^\alpha}{x^{\alpha+1}}, \quad x > \beta, \quad \text{mean} = \frac{\alpha \beta}{\alpha - 1} \quad (\text{for } \alpha > 1).$$

Furthermore, as a simplification for the simulations, the statistics of the event absent times Y_k^i 's is taken to be the same as that for the event staying times X_k^i 's, even though this is by no means necessary.

We first explain the intuition in analyzing the QoM function. The main step in computing the QoM at a PoI is to consider the overlapping periods during which both the event and the sensor are present at the same PoI. A complication is that the sensor can leave and come back multiple times to the same PoI and observe the same event. Hence, the *total* observation period of a single event will in general be a collection of *disjoint* time intervals. See Figure 3.

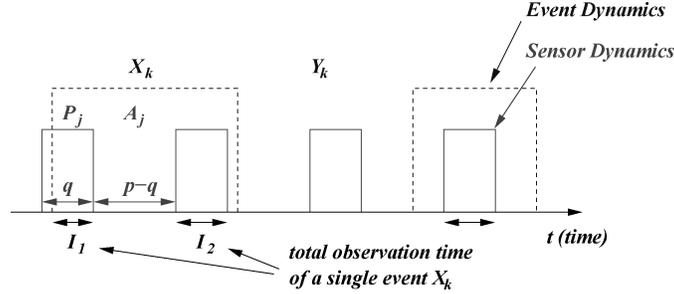


Fig. 3. Event and sensor dynamics at a PoI. The X_k 's and Y_k 's are the event present and absent periods and the P_j 's and A_j 's are the sensor present and absent periods. The lengths of the P_j 's and A_j 's equal q and $p - q$ respectively. In the above example, the total observation period of the event X_k is the sum of I_1 and I_2 .

For the convenience of the presentation that follows, we denote the proportional share $\frac{q}{p}$ by γ . Furthermore, we use $P_j = [(j - 1)p, (j - 1)p + q]$ and $A_j = [(j - 1)p + q, jp]$ to refer to the j th sensor present and absent periods, respectively. For many of the proofs, it is sufficient to consider just the case $j = 1$, that is, $P_1 = [0, q]$ and $A_1 = [q, p]$. This is illustrated in Figure 3.

As an example, to compute the utility acquired of the event X_k^i in Figure 3, note that the total observation time of the event equals $I_1 + I_2$. Hence the utility is given by $U(I_1 + I_2)$. This is analytically computable as the statistics of X_k^i is assumed to be known and the sensor movement is periodic. The machinery of renewal theory is used to handle the statistics of the starting point of the event.

We gradually establish our results and understanding by first considering the step utility function with blip and staying events. Then we write down formulas for general utility functions. Several analytic results are obtained for events with Exponential and Pareto distributions. Now we proceed to present our results.

4.1 Step Utility Function

We begin our discussion with events that have the step utility function $U_I(x)$ (see Figure 2). In this case, since the utility reaches one instantaneously, the QoM is equivalent to the fraction of events captured. The next result illustrates the effect on the QoM by a periodic sensor schedule with parameters p and q at a fixed PoI.

THEOREM 4.1. *For independent arrivals of events that have the step utility function and do not stay, that is, “blip events,” the QoM at any PoI is directly proportional to its share of coverage time q/p . In particular, the achieved QoM does not depend on the fairness granularity.*

PROOF. The statement is a simple consequence of the fact that an event is completely captured if and only if it arrives when the sensor is present. Hence,

the QoM is simply the ratio between the expected number (per unit time) of arrivals during the sensor present period q and the total period p , that is, $\frac{q}{p}$. \square

The above scenario shows that only the proportional sharing information determines the QoM. On the other hand, for events that do stay, the QoM depends on the relationship between the event staying time distribution and the parameters p and q . Specifically, we have the following result.

THEOREM 4.2. *For independent arrivals of events that stay and have the step utility function, the QoM at a PoI is given by*

$$Q = \frac{q}{p} + \frac{1}{p} \int_0^{p-q} \Pr(X \geq t) dt. \quad (3)$$

PROOF. As the utility function is a step function, the overall utility is given by the total number of events captured when the sensor is present. Note that an event will be captured if (i) it arrives during the sensor present period $[0, q]$; (ii) it arrives during the sensor absent period $[q, p]$, but stays long enough to be captured during the *next* sensor present period $[p, p+q]$. The contribution of (i) to the QoM is given by $\frac{q}{p}$, while that of (ii) is given by $\frac{1}{p} \int_q^p \Pr(X+t \geq p) dt$, which is the second term of Eq. (3) after a simple change of variable. \square

Theorem 4.2 implies that the sensor that stays at a PoI for $\gamma = q/p$ fraction of the time may be able to capture a significantly *larger* fraction of events than q/p . The following two corollaries give further statements about this *extra* fraction of events.

COROLLARY 4.3. *Under the setting of Theorem 4.2, with the fairness granularity p kept constant, we have:*

$$\lim_{\gamma \rightarrow 0} Q = \frac{1}{p} \int_0^p \Pr(X \geq t) dt. \quad (4)$$

PROOF. The proof is a direct consequence of Eq. (3), upon taking the limit $\gamma \rightarrow 0$. (Note that $q = \gamma p \rightarrow 0$.) \square

This result clearly indicates that no matter how small the proportional share is, there is always some definite, positive gain of information. This is due to the fact that the events stay.

COROLLARY 4.4. *Under the setting of Theorem 4.2, the QoM of a given fixed proportional share γ is a monotonically decreasing function of the fairness granularity, that is, Q decreases as p increases. Furthermore,*

$$\lim_{p \rightarrow 0} Q(p) = 1, \quad \text{and} \quad \lim_{p \rightarrow \infty} Q(p) = \frac{q}{p}. \quad (5)$$

PROOF. The statement again is a simple consequence of Eq. (3), which is rewritten in the following form:

$$Q = \gamma + (1 - \gamma) \frac{1}{(1 - \gamma)p} \int_0^{(1-\gamma)p} \Pr(X \geq t) dt.$$

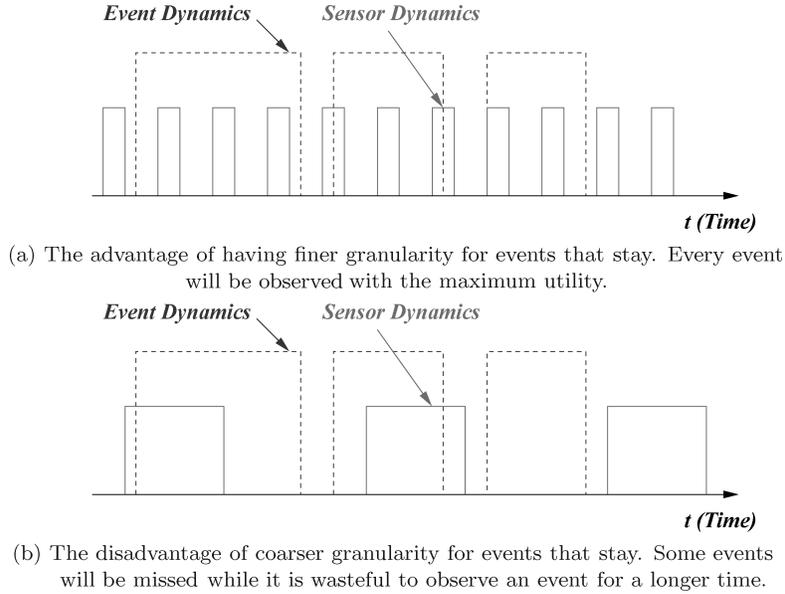


Fig. 4. Step utility function. Illustration of sensor dynamics versus event dynamics.

Note that the second term in this equation is the average over the interval $[0, (1 - \gamma)p]$ of the monotonically decreasing function $\Pr(X \geq t)$ of t . Furthermore, $\lim_{t \rightarrow 0} \Pr(X \geq t) = 1$ and $\lim_{t \rightarrow \infty} \Pr(X \geq t) = 0$. Hence,

$$\lim_{p \rightarrow 0} \frac{1}{(1 - \gamma)p} \int_0^{(1 - \gamma)p} \Pr(X \geq t) dt = 1$$

and

$$\lim_{p \rightarrow \infty} \frac{1}{(1 - \gamma)p} \int_0^{(1 - \gamma)p} \Pr(X \geq t) dt = 0,$$

which leads to the stated result. \square

In contrast to Theorem 4.1, which applies for blip events, Corollary 4.4 implies that finer-grained fairness *does* generally improve the QoM for events that stay and have the Step utility function. In particular, no matter how small the proportional share is, an arbitrarily high QoM can be achieved by an extremely fine fairness granularity. This makes intuitive sense for two reasons. First, by leaving and coming back to a PoI *infinitely often* and *fast*, the sensor can capture any event (which stays) while maintaining the desired proportional sharing. Second, if the granularity is coarse, there is a definite amount of time during which the sensor is absent. Thus, some events will be missed while those events which are already observed will not lead to a higher QoM as the *maximum* amount of utility is already achieved by the *first* moment the event is observed. Hence, it is advantageous to leave the PoI and search for *other new* events. The above reasoning is illustrated in Figure 4.

The following are some explicit examples to illustrate Theorem 4.2 and Corollary 4.4.

(1) *Exponential Distribution.*

$$Q = \gamma + \frac{1}{p} \int_0^{(1-\gamma)p} \int_t^\infty \lambda e^{-\lambda x} dx dt = \gamma + \frac{1 - e^{-\lambda(1-\gamma)p}}{\lambda p}, \quad (6)$$

which converges to 1 and γ as $p \rightarrow 0$ and ∞ .

(2) *Pareto Distribution.* When $(1-\gamma)p \leq \beta$, then $Q = 1$ because any event will always be captured as its duration is at least β time units long. Hence,

$$Q = \gamma + \frac{1}{p} \int_0^{(1-\gamma)p} \Pr(X \geq t) dt = 1. \quad (7)$$

When $(1-\gamma)p > \beta$, then Q is given by

$$Q = \gamma + \frac{1}{p} \left[\int_0^\beta \Pr(X \geq t) dt + \int_\beta^\infty (1-\gamma)p \Pr(X \geq t) dt \right],$$

which equals

$$\gamma + \frac{1}{p} \left[\beta + \frac{\beta^\alpha}{(\alpha-1)} \left(\frac{1}{\beta^{\alpha-1}} - \frac{1}{((1-\gamma)p)^{\alpha-1}} \right) \right]. \quad (8)$$

The preceding expression also converges to γ as $p \rightarrow \infty$.

We now consider a scaling result for mobile sensor coverage k out of n PoIs, whose event arrival and departure processes are i.i.d., as k increases. Assume that initially, the sensor performs periodic schedules among k of the n PoIs such that $q^i = \delta$ and $p^i = k\delta$, for $1 \leq i \leq k$, where δ is a unit of time. The following theorem holds.

THEOREM 4.5. *The expected fraction of events captured is an increasing function of k , the number of PoIs covered.*

PROOF. In accordance with (2), the overall QoM is given by:

$$\begin{aligned} Q_* &= \frac{1}{n} \sum_{1 \leq j \leq k} Q_j = \frac{1}{n} \sum_{1 \leq j \leq k} \left[\frac{1}{k} + \frac{1}{k\delta} \int_0^{(k-1)\delta} P(X \geq t) dt \right] \\ &= \frac{1}{n} \left[1 + \frac{1}{\delta} \int_0^{(k-1)\delta} P(X \geq t) dt \right] \end{aligned}$$

which is clearly an increasing function of k . \square

Theorem 4.5 provides a formal justification for mobile coverage, namely that the amount of information captured increases as the sensor moves among more PoIs to search for interesting information. This is in addition to the obvious advantage of fairly distributing the sensing resources among the PoIs.

4.2 General Utility Function

We now turn our attention to events that have a general utility function $U(\cdot)$. In this case, we have the following QoM result.

THEOREM 4.6. *For independent arrivals of events at a PoI that have the utility function $U(\cdot)$ and whose event staying time pdf is given by $f(x)$, the*

achieved QoM equals ($\xi_i = iq - t$, $\eta_i = x + ip - t$):

$$\int_0^q \left[\int_0^{q-t} U(x) f(x) dx + \sum_{i=1}^{\infty} \int_0^q U(\xi_i + x) f(\eta_i) dx + \sum_{i=1}^{\infty} U(\xi_i) \int_{-(p-q)}^0 f(\eta_i) dx \right] dt \quad (9)$$

$$+ \int_q^p \left[\sum_{i=1}^{\infty} \int_0^q U(\xi_i - q) f(\eta_i) dx + \sum_{i=1}^{\infty} U(\xi_i + t) \int_q^p f(\eta_i) dx \right] dt. \quad (10)$$

PROOF. The above formula follows from the fact that the overall utility available for any particular event depends on the *total* length of the intersecting region (which might be discontinuous) during which both the event and sensor are present. The various summands in integral (9) and (10) correspond to the cases that the event arrives when the sensor is present or absent.

If an event arrives at $t \in [0, q]$, that is, when the sensor is present, then the total utility available from this event is given by ($\xi_i = iq - t$):

$$\int_0^{q-t} U(x) f(x) dx + \sum_{i=1}^{\infty} \int_{x+t=ip}^{x+t=ip+q} U(\xi_i + x + t - ip) f(x) dx + \sum_{i=1}^{\infty} \int_{x+t=ip-(p-q)}^{x+t=ip} U(\xi_i) f(x) dx.$$

In the above, the different integrals correspond to the cases when the event departure time $t + x$ falls in $[t, q]$, $[ip, ip + q]$, and $[ip - (p - q), ip]$, respectively. A change of variable gives (9).

Similarly, if an event arrives at $t \in [q, p]$, that is, when the sensor is absent, then the total utility available from this event is given by:

$$\sum_{i=1}^{\infty} \int_{x+t=ip}^{x+t=ip+q} U((i-1)q + x + t - ip) f(x) dx + \sum_{i=1}^{\infty} \int_{x+t=ip-(p-q)}^{x+t=ip} U((i-1)q) f(x) dx.$$

A change of variable formula then also gives the form of (10). \square

Formula (9) and (10) can have a complicated analytical form in general, but they are certainly amenable to numerical computation. Nevertheless, we first present two exact analytical results. (Recall $\gamma = \frac{q}{p}$.)

(1) *Exponential Utility Function* $U_E(x) = 1 - e^{-Ax}$ and *Exponential Staying Time*: $f(x) = \lambda e^{-\lambda x}$.

$$\begin{aligned}
 Q &= \frac{A\gamma}{A+\lambda} - \frac{1 - e^{-\lambda q}}{\lambda p} + \frac{\lambda(1 - e^{-(A+\lambda)q})}{(A+\lambda)^2 p} \\
 &+ \frac{(e^{\lambda q} - 1)^2}{\lambda p e^{\lambda q} (e^{\lambda p} - 1)} - \frac{\lambda(e^{(A+\lambda)q} - 1)^2 e^{-(A+\lambda)q}}{(A+\lambda)^2 p (e^{(Aq+\lambda p)} - 1)} \\
 &+ \frac{2(e^{\lambda(p-q)} - 1)}{p} \times \left[\frac{e^{\lambda q} - 1}{\lambda(e^{\lambda p} - 1)} - \frac{e^{(A+\lambda)q} - 1}{(A+\lambda)(e^{(Aq+\lambda p)} - 1)} \right] \\
 &+ \frac{(e^{Aq} - 1)e^{\lambda q}(e^{\lambda(p-q)} - 1)^2}{\lambda p (e^{\lambda p} - 1)(e^{(Aq+\lambda p)} - 1)}. \tag{11}
 \end{aligned}$$

Note that this equation leads to

$$\lim_{p \rightarrow 0} Q = \frac{A\gamma}{A\gamma + \lambda}, \quad \text{and} \quad \lim_{p \rightarrow \infty} Q = \frac{A\gamma}{A + \lambda}. \tag{12}$$

(2) *Delayed Utility Function* $U_D(x) = U_I(x - D)$ and *Exponential Staying Time*: $f(x) = \lambda e^{-\lambda x}$.

When p is very small such that D is an integral multiple of q , that is, $D = kq$ for $k = 1, 2, \dots$, we have:

$$Q = e^{-\frac{\lambda D}{\gamma}} \left[\gamma + \frac{e^{\lambda(1-\gamma)p} - 1}{\lambda p} \right]. \tag{13}$$

On the other hand, when p is very large, specifically, when $q > D$, then

$$Q = e^{-\lambda D} \left[\gamma + \left(\frac{1}{\lambda} - D \right) \frac{1 - e^{-\lambda(p-q)}}{p} \right]. \tag{14}$$

It is also interesting to obtain Eq. (14) without using Theorem 4.1. In order for the sensor to capture an event and gain enough information, the event staying time and the sensor present period should overlap for at least an interval of length D . Based on this, let t be the time that an event occurs and let x be its staying time. Consider the first sensor present and absent periods $[0, q]$ and $[q, p]$. Then, the probability of gaining enough information for an event arriving during these two periods is given by

$$\begin{aligned}
 &\int_0^{q-D} \Pr(X \geq D) dt \\
 &+ \int_{q-D}^q \Pr(t + x - p + (q - t) \geq D) dt \\
 &+ \int_q^p \Pr(t + x - p \geq D) dt,
 \end{aligned}$$

which gives the result (14) upon dividing by p .

Combining Eq. (13) and (14), we have:

$$\lim_{p \rightarrow 0} Q = e^{-\lambda \frac{D}{\gamma}}, \quad \text{and} \quad \lim_{p \rightarrow \infty} Q = \gamma e^{-\lambda D}. \quad (15)$$

For the Pareto type events, even though analogous results can be derived, the analytical formulas are quite complicated and hard to be put into closed form. But they demonstrate similar qualitative behaviors as seen in the simulation section.

These analytical results can be intuitively understood in many ways, which are instructive to discuss.

4.3 Implications of Theoretical Results

The first three discussion points concern various limiting cases.

- (i) Let the fairness granularity p and the proportional share γ be fixed. Then as the event staying time goes to infinity, every event will always be captured and the maximum value 1 for the utility can be achieved. Therefore, the QoM is an increasing function of the mean event staying time. Note that this scenario corresponds to $\lambda \rightarrow 0$ for the exponential staying time distribution, and $\beta \rightarrow \infty$ for the Pareto distribution.
- (ii) In the limit of $p \rightarrow 0$, every event which stays will always be captured. However, the total observation time is only γ fraction of the event's duration. Hence, the average utility achieved is:

$$Q_0 = \int_0^\infty U(\gamma x) f(x) dx. \quad (16)$$

This result is consistent with the explicit results (12) and (15).

The following give an explicit expression of the QoM for the Pareto event staying time distribution.

—With the exponential utility function U_E ,

$$Q_0 = 1 - \alpha \beta^\alpha \int_\beta^\infty \frac{e^{-A\gamma x}}{x^{\alpha+1}} dx = 1 - \alpha \int_1^\infty \frac{e^{-A\gamma\beta x}}{x^{\alpha+1}} dx.$$

—With the Delayed utility function U_D ,

$$Q_0 = \begin{cases} 1 & \text{for } D \leq \gamma\beta, \\ \left(\frac{\gamma\beta}{D}\right)^\alpha & \text{for } D > \gamma\beta. \end{cases}$$

- (iii) In the limit of $p \rightarrow \infty$, each event, if captured, will essentially be observed for its whole duration. On the other hand, only γ fraction of the events will be captured. Hence, the QoM is given by:

$$Q_\infty = \gamma \int_0^\infty U(x) f(x) dx, \quad (17)$$

which is also consistent with the explicit results (12) and (15).

Again, for Pareto event staying time distribution, we have:

—with the Exponential utility function:

$$Q_\infty = \gamma \left[1 - \alpha \beta^\alpha \int_\beta^\infty \frac{e^{-Ax}}{x^{\alpha+1}} dx \right] = \gamma \left[1 - \alpha \int_1^\infty \frac{e^{-A\beta x}}{x^{\alpha+1}} dx \right].$$

—with the Delayed utility function:

$$Q_\infty = \begin{cases} \gamma & \text{for } D \leq \beta, \\ \gamma \left(\frac{\beta}{D}\right)^\alpha & \text{for } D > \beta. \end{cases}$$

The next two discussion points concern the two most important qualitative descriptions of the QoM function.

- (iv) For the step and exponential utility functions, the QoMs are monotonically decreasing functions of p . This is because both utility functions are concave functions of the observation time. Hence, it is advantageous to capture as many new events as possible rather than to gain further information for the event which has already been observed. A finer fairness granularity exactly achieves this. More precisely, a very fine granularity can basically capture *every* single event (as the event stays), and each event captured gives the best possible initial utility gain *per unit time*. On the other hand, coarser granularity can miss some events while for those events captured, the information captured per unit time is not maximized due to the concavity of the utility function.

This function is certainly consistent with Theorem 4.2 for the step utility function, which is qualitatively illustrated in Figure 4. Furthermore, This is also explicitly demonstrated by the analytical formula (6), (7)–(8). It is easy to see that their derivatives with respect to p is negative. For (11), this behavior is graphically demonstrated in Figure 5.

- (v) However, the key feature is that for certain utility functions, the maximum QoM is only achieved at some intermediate fairness granularity. We spend a moment to explain this important phenomenon.

This observation is easiest to explain for the delayed step utility U_D . In the limit of $p \rightarrow 0$, any event can always be captured. This is essentially the statement of Corollary 4.4. However, in order to gain enough information about the event, it is necessary that the event staying time be at least $\frac{D}{\gamma}$ long. This probability is given by $\Pr(X \geq \frac{D}{\gamma})$. However, when p is positive (no matter how small it is), this is not absolutely necessary. In fact, if the event arrives right at the beginning of a sensor present period, then the event staying time just needs to be at least $\frac{D}{\gamma} - (1 - \gamma)p$ long. It is this saving that increases the QoM. Hence, initially, the QoM is an *increasing* function of p for *small* p . This can also be seen analytically from Eq. (13) by which we have for $0 < p \ll 1$:

$$\text{QoM} \approx e^{-\frac{\lambda D}{\gamma}} \left[1 + \frac{(1 - \gamma)^2 \lambda p}{2} + \dots \right]$$

which is an increasing function of p .

The behavior of QoM when p is large is also interesting and in fact quite intricate. From Eq. (14), observe that the QoM is a decreasing, constant, or increasing function of p for λ less than, equal to, or greater than $\frac{1}{D}$, respectively. This can be seen that for $q \gg 1$, we have:

$$\frac{d\text{QoM}}{dp} \approx e^{-\lambda D} \left[\gamma - \left(\frac{1}{\lambda} - D \right) \frac{1}{p^2} + \dots \right]$$

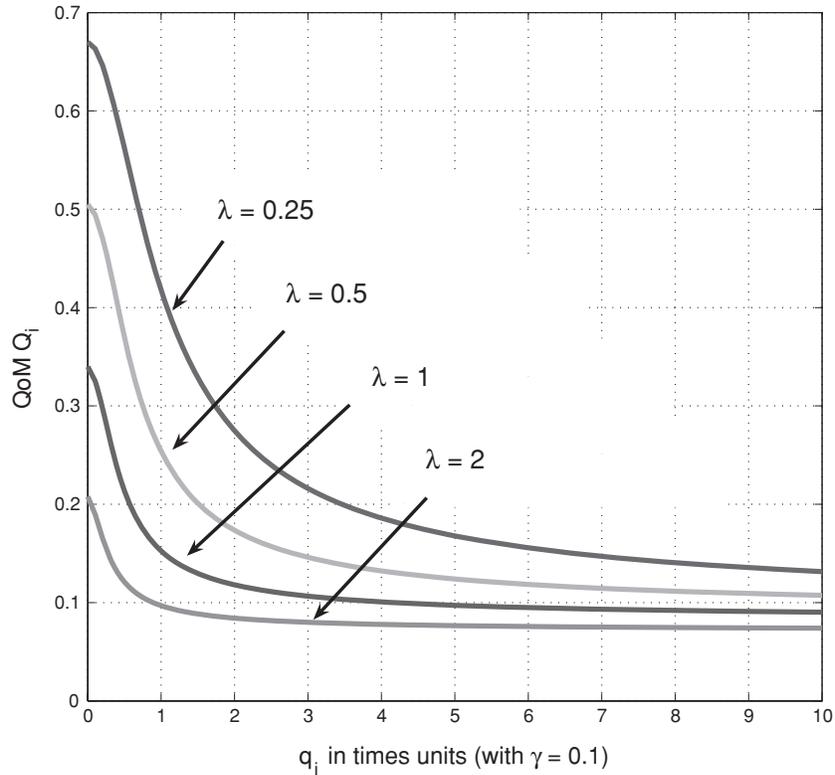


Fig. 5. The QoM for the exponential utility function and exponential staying time.

This equation is due to the competitive effect (for p large) of the loss of utility for events arriving near the end of a sensor present period and the gain of utility for events arriving before the sensor present period. Hence, for $\lambda < \frac{1}{D}$, the QoM initially increases and then decreases as a function of p . Thus, it is optimal at some intermediate p value.

All of the preceding implications are supported by the simulation results in Section 6.

4.4 Discussions on Multiple Sensors

In the previous analysis, it is simplest to interpret that the periodic PoI visit schedule is induced by a single periodic sensor. We now discuss an extended scenario in which multiple mobile sensors, each on the same periodic schedule, visit the PoI in sequence. We acknowledge that the general coordination between multiple sensors is a very important problem of practical and research relevance. A full analysis of the scenario deserves another line of work and is out of the scope of the current article. On the other hand, we will demonstrate that the present QoM analysis can already lead to some interesting consequences.

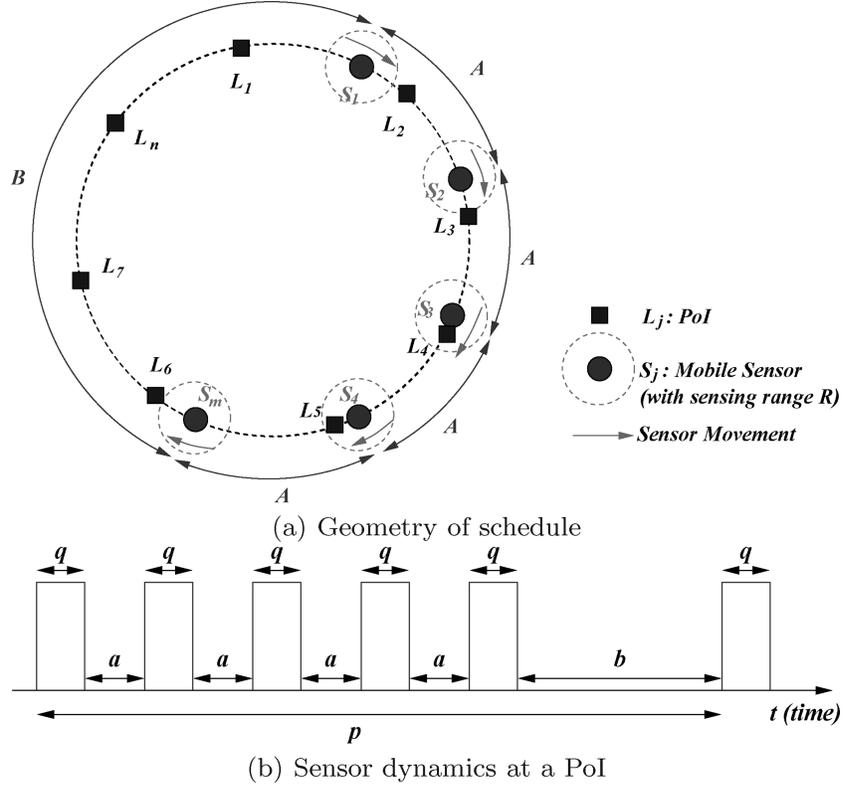


Fig. 6. Geometry of schedule and sensor dynamics at a PoI using multiple sensors.

We assume that the PoIs $\{L_i\}_{i=1,2,\dots,n}$'s are located along a circular circuit. The mobile sensors $\{P_j\}_{j=1,2,\dots,m}$ move along the circuit in an identical fashion. The geometry of the sensor locations is such that they are clustered together in the following sense (see Figure 6(a)).

$$\begin{aligned} \text{dist}(P_j, P_{j+1}) &= A, \quad \text{for } j = 1, 2, \dots, m-1 \\ \text{dist}(P_m, P_1) &= B \end{aligned}$$

The distance is measured in the clockwise sense. From each PoI's point of view, the pattern of the coverage time is illustrated in Figure 6(b). As each sensor is associated with a sensing range R , the coverage time is finite, denoted by q . In addition, due to the separations A and B between the sensors, we let the sensor absent times be a and b . We investigate the QoM for each PoI in relation to the following quantities.

- (1) p : the total period of the sensor movement
- (2) m : the number of mobile sensors
- (3) $r = \frac{b}{a}$: the "clustering ratio" of the sensors' visit sequence

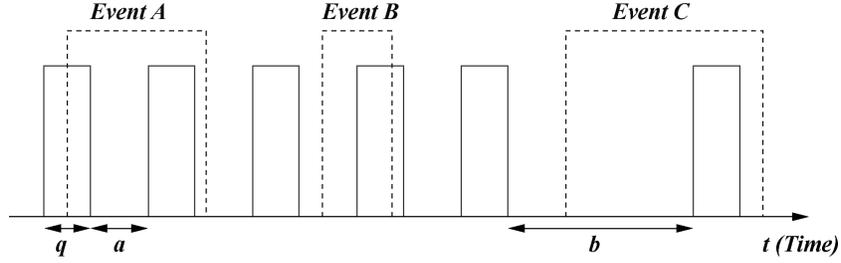


Fig. 7. The relation between event dynamics and the multisensor coverage pattern. In this figure, the number of sensors m equals 5.

Note the relation $p = mq + (m - 1)a + b$. In addition, $r = 1$ implies that the sensors are equidistant from each other while for $r = \infty$, it is equivalent to having a single mobile sensor with coverage time $q' = mq$ for each period p . Furthermore, the information acquired by the sensors is aggregated for possibly higher combined utilities.

First, we consider the case of the step utility function. In this case, any event will be captured with the maximum utility value one if it occurs when the sensor is present (see Figure 7, event A). On the other hand, if it occurs when the sensor is absent, then the event must stay at least till the first moment the sensor comes back in order to be captured (see Figure 7, events B and C). Based on this argument, the QoM function is given by:

$$\text{QoM} = \frac{1}{p} \left[m \int_0^q dt + (m - 1) \int_0^a P(T \geq a - t) dt + \int_0^b P(T \geq b - t) dt \right] \quad (18)$$

where T is the event staying time. The prefactors before the integrals in this expression come from the fact that there are m identical cases of the event type A, $m - 1$ identical cases of type B, and one case of type C.

To give further explicit analysis, we assume that the event staying time follows the exponential distribution with parameter λ . In this case Eq. (18) is reduced to:

$$\text{QoM} = \frac{1}{p} \left[mq + \frac{m - (m - 1)e^{-\lambda a} - e^{-\lambda b}}{\lambda} \right]. \quad (19)$$

In addition, by means of $p = mq + (m - 1)a + b$ and $r = \frac{b}{a}$, we get $a = \frac{p - mq}{m - 1 + r}$. Hence, the above QoM function becomes:

$$\text{QoM} = \frac{1}{p} \left[mq + \frac{m - (m - 1)e^{-\lambda \left(\frac{p - mq}{m - 1 + r} \right)} - e^{-\lambda r \left(\frac{p - mq}{m - 1 + r} \right)}}{\lambda} \right]. \quad (20)$$

This QoM function is plotted in Figure 8.

The monotonically decreasing behavior is expected as explained previously. It can also be seen by taking the derivative of the QoM with respect to the clustering ratio r . The result is given by:

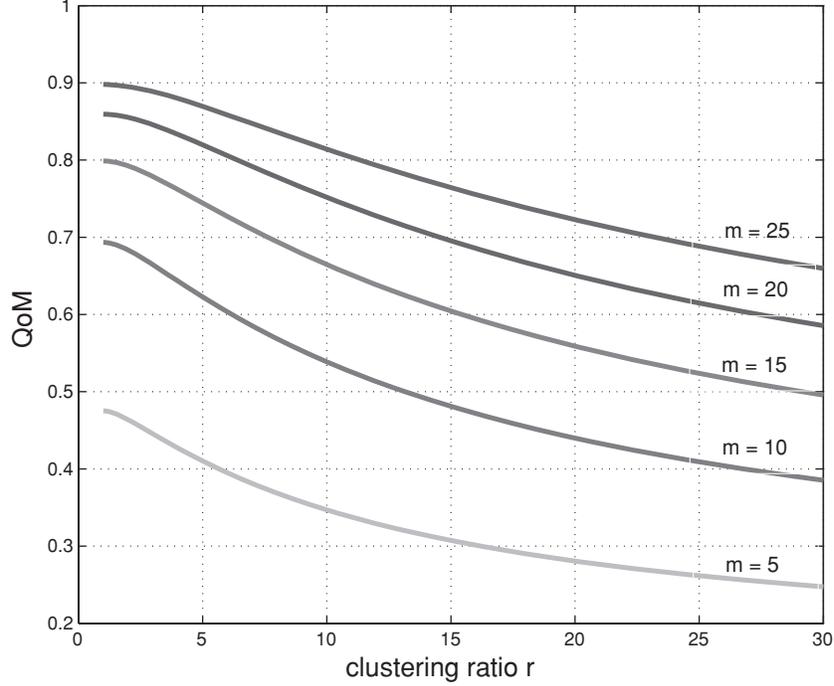


Fig. 8. Plot of the QoM function (20) for the step utility function: $p = 10$, $q = 0.1$, and varying number of mobile sensors m .

$$\frac{d\text{QoM}}{dr} = \frac{(m-1)(p-mq)}{p(m-1+r)^2} \left[-e^{-\lambda\left(\frac{p-mq}{m-1+r}\right)} + e^{-\lambda r\left(\frac{p-mq}{m-1+r}\right)} \right], \quad (21)$$

which is clearly negative for $r \geq 1$.

We now consider the second, more interesting case of the delayed step utility function with delay D . The argument is similar to the previous case, except that in order to gain the maximum utility value one, the event must now stay long enough so that the total observation time is at least D .

To obtain a tractable mathematical formula, we assume that $D = Nq$ for some positive integer N . Then, based on the consideration of different event types such as A , B , and C in Figure 7, the QoM function is given by:

$$\begin{aligned} \text{QoM} = & \frac{1}{p} \left[(m-N) \int_0^q P(T \geq (a+q)N) dt + (m-N) \int_0^q P(T \geq (a+q)N-t) dt \right. \\ & + N \int_0^q P(T \geq (a+q)N-a+b) dt + (N-1) \int_0^a P(T \geq (a+q)N-a+b-t) dt \\ & \left. + \int_0^b P(T \geq (a+q)N-a+b-t) dt \right]. \end{aligned}$$

As before, by assuming the exponential distribution for the event staying time, we have

$$\begin{aligned} \text{QoM} = \frac{1}{p} \left\{ (m - N)e^{-\lambda(\alpha+q)N} \left[q + \frac{e^{\lambda\alpha} - 1}{\lambda} \right] \right. \\ \left. + e^{-\lambda(\alpha+q)N+b-a} \left[Nq + (N - 1)\frac{e^{\lambda\alpha} - 1}{\lambda} + \frac{e^{\lambda b} - 1}{\lambda} \right] \right\} \quad (22) \end{aligned}$$

Function (22) is plotted in Figure 9.

From the above, it appears that the behavior of the QoM is quite elaborate as a function of the clustering ratio r and the number of mobile sensors m . In particular, it can be a monotonically increasing, nonmonotonic, or monotonically decreasing function of m , depending on the *competition* between eliminating redundancies in the information capture and avoiding the *loss* of events. See also the explicit discussion in Section 4.3(v).

5. COVERAGE ALGORITHMS

The previous section discussed the QoM of periodic schedules at a specific single PoI. We now address the problem of covering n PoIs by the sensor. This is achieved by a visit schedule of the sensor to all the PoIs under a coverage algorithm to be designed.

We will analyze the QoM of *periodic* coverage of n PoIs. By this we mean that the schedule is realized by a periodic visit schedule of the sensor to the PoIs, in which the visit schedule in the smallest period is denoted by

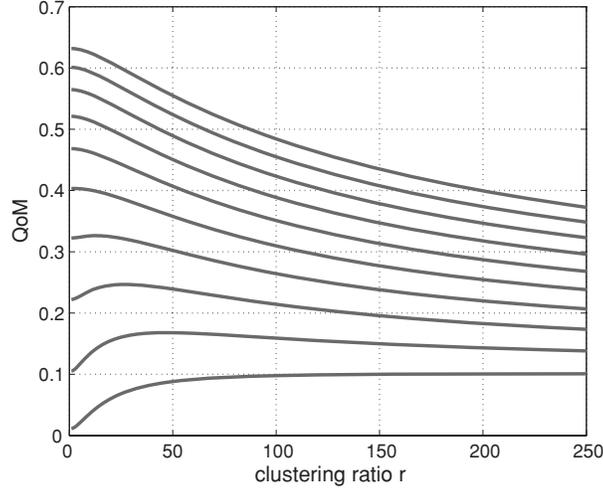
$$S = \{(L_1, C_1), \dots, (L_l, C_l)\}, \quad (23)$$

where L_j denotes the j th PoI visited for a time of C_j in the sensor schedule, $L_j \neq L_{(j \bmod l)+1}$, and each of the n distinct PoIs appears at least once in S . Recall from Assumption 2 on Page 7 that the sensor cannot be present at more than one PoI at a time. If $l = n$, that is, each PoI appears in S exactly once, then we call S a *linear periodic schedule*. However, it is clear that not all periodic schedules are linear. For example, $S = \{(1, \delta), (2, 3\delta), (1, \delta), (3, 2\delta)\}$, where δ is a unit of time, is not. In Definition (23), if $l > n$, we call the periodic schedule *nonlinear*.

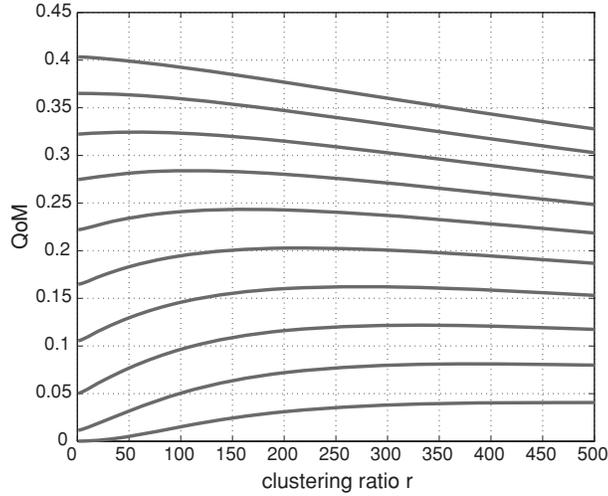
Next, we perform analysis of linear periodic schedules. The case of general nonlinear schedule comes afterwards. Given a sensor schedule S , with prescribed PoI coverage times C_i 's, we define its *maximum feasible utilization* as

$$U_*(S) = \sup \left\{ \sum_{1 \leq i \leq n} \frac{q^i}{p^i} \right\},$$

where the sup is taken over all possible sensor movements that realize S . This utilization is affected by the travel time overhead between two adjacent PoIs in S during which the sensor is not present at any PoI. Using $d(i, j)$ as



(a) $p = 10$, $q = 0.01$, $D = 0.05$. The curves (from below to above) corresponds to $m = 10, 20, 30, \dots, 100$.



(b) $p = 100$, $q = 0.01$, $D = 0.05$. The curves (from below to above) corresponds to $m = 50, 100, 150, \dots, 500$.

Fig. 9. Plot of the QoM function (22).

an equivalent notation to d_{ij} for the distance between i and j , we define for $j = 1, \dots, l$:

$$a_j = \frac{1}{v_{\max}} \left[d(L_j, L_{(j \bmod l)+1}) - 2R \right]$$

as the minimum travel time overhead from L_j to $L_{(j \bmod l)+1}$ for the sensor moving at maximum speed v_{\max} . This geometric setting is illustrated in Figure 10.

With this setting, we have the following statement.

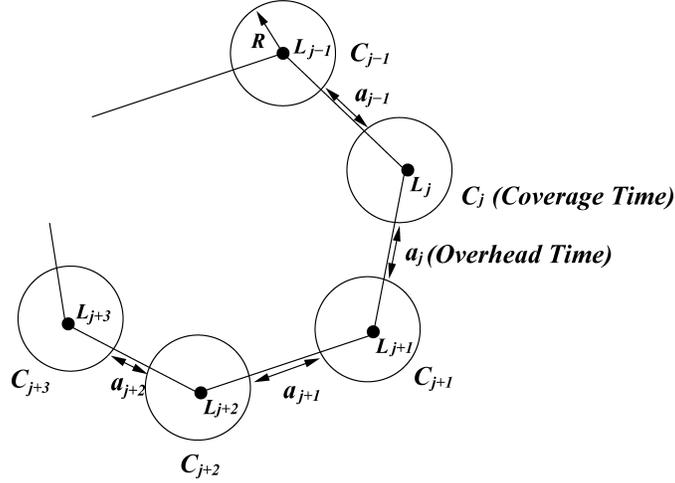


Fig. 10. Geometry of linear scheduling. The L_j 's denote the PoIs. R is the range of the mobile sensor. The C_j 's and a_j 's are the coverage times of the PoIs by the sensor and the overheads when the sensor is out of range of any PoIs.

THEOREM 5.1 (LINEAR PERIODIC SCHEDULE). *The maximum feasible utilization of S is*

$$U_*(S) = \sup \left[1 - \frac{\sum_{1 \leq j \leq l} a_j}{\sum_{1 \leq j \leq l} (C_j + a_j)} \right],$$

where the sup is taken over all possible sensor movements realizing S .

PROOF. Note that completing one period of the sensor schedule requires $P_* = \sum_{1 \leq j \leq l} (C_j + a_j)$ time units. As the schedule is linear, all the p^i 's equal P_* . Hence, the proportional share for PoI i is given by $\frac{C_j}{P_*}$. The result thus follows from: $\sum_j \frac{q^j}{p^j} = \sum_j \frac{C_j}{P_*} = 1 - \frac{1}{P_*} \sum_j a_j$. \square

Theorem 5.1 shows that 100% sensor utilization is feasible if and only if each adjacent pair of PoIs in S are exactly $2R$ apart. In actual application, we would like to maximize $U_*(S)$. As its form is a decreasing function of the sum $\sum_{i \leq j \leq l} a_j$, we would indeed want to minimize the travel overhead.

5.1 Optimization of Linear Periodic Schedule

Here we discuss the optimization of the QoM Q_* (defined in Section 2) for the overall system in the realm of linear periodic schedules. The solution must satisfy a given proportional fairness objective, that is, for each pair of PoIs, say i and j , we must achieve a given ratio γ_{ij} for their shares of coverage time, that is, for the periodic schedules induced by S , we must have $\frac{q^i/p^i}{q^j/p^j} = \gamma_{ij}$ for all i and j 's.

A linear periodic schedule exists if there is a Hamiltonian circuit of the PoIs. An optimization approach for linear periodic schedules works as follows. We

first determine the visit order of the PoIs in S that will minimize $\sum_{1 \leq j \leq l} a_j$. The problem is the Traveling Salesman Problem and is NP hard, but practical approaches exist that give solutions within a few percent of the optimal for problem sizes of up to 100,000 [Kirkpatrick et al. 1983; Cerny 1985]. Once the visit order is determined, a_j , $j = 1, \dots, l$, is known, and it remains to determine the C_j , $j = 1, \dots, l$. Notice that in a linear periodic schedule, $l = n$, $C_j = q^j$, and $p^1 = \dots = p^n = \sum_j (C_j + a_j) = P_*$. We first select each C_j to satisfy $C_j = \gamma_{j1} C_1$ so that all the coverage times can be expressed in terms of C_1 only. This greatly simplifies the problem as it becomes a purely one-dimensional optimization problem. The choice of C_1 that optimizes Q_* depends on the event utility function U . We illustrate the above approach by a simple example.

First, consider first blip events and the step utility function U_I . If $\sum_j a_j = 0$, then any choice of C_1 is optimal as the QoM is simply the fraction of events captured at the PoIs. More precisely,

$$Q_* = \frac{1}{nP_*} \sum_j C_j = \frac{1}{n}.$$

On the other hand, if $\sum_j a_j > 0$, then the optimal Q_* cannot be attained but it can be approached as closely as possible by using a finite but sufficiently large value of C_1 .

Second, for general event utility functions, we need to compute the corresponding QoM Q_i for each i using Theorem 4.6. Recall that $C_i = \gamma_{i1} C_1$, and Q_* is expressible as a weighted sum of the individual Q_i 's (from Eq. 2):

$$Q_* = \frac{1}{\mu_*} \sum_j \mu_j Q_i \left(\frac{\gamma_{j1} C_1}{P_*} \right).$$

Therefore Q_* is a function of C_1 only. The value of C_1 that optimizes QoM Q_* can be computed by solving

$$\frac{dQ_*}{dC_1} = 0, \quad \text{and} \quad \frac{d^2 Q_*}{dC_1^2} < 0.$$

Note that Q_* can possibly have multiple local maxima as each Q_i has its own optimal C_i 's. But the issue can be easily resolved by a numerical search since the problem is one-dimensional.

5.2 General Nonlinear Periodic Coverage

The previous section discussed the optimization of linear periodic sensor schedules. However, a linear periodic schedule does not exist if there is no Hamiltonian circuit of the PoIs. Even if it exists, a linear schedule is in general sub-optimal as the QoM depends on the fairness granularity (Corollary 4.4). This is illustrated by the following example. Consider three PoIs, located such that $d_{12} = d_{13} = d_{23} = 2R$, and the proportional fairness objective of $\gamma_{12} = l/(l-1)$ and $\gamma_{13} = l$. For events that stay and have the step utility function, the optimal linear periodic sensor schedule is $\{(1, l\delta), (2, (l-1)\delta), (3, \delta)\}$, where $\delta = 2R/v_{\max}$

is the minimum presence time of the sensor arriving at and then leaving a PoI—recall Assumption 1. From Theorem 4.2, however, we know that the QoM at i increases as the fairness granularity decreases. Hence, the following optimal *nonlinear periodic* schedule:

$$\underbrace{\{(1, \delta), (2, \delta), \dots, (1, \delta), (2, \delta), (1, \delta), (3, \delta)\}}_{l-1 \text{ times}}$$

increases the QoM at 1 and 2 without affecting either the travel overhead or the QoM at 3. When l is large, the performance loss of the optimal linear schedule can be significant for certain distributions of the event staying time, for example, when the mean event staying time is on the order of δ .

The significant performance loss of linear periodic schedules argues for the need to search for general periodic schedules with better performance. A beginning observation is that a new and potentially better periodic schedule can be obtained by rearranging the PoI order in an original schedule. Changing the PoI order affects the fairness granularity as discussed above, but it also affects the travel overhead between the adjacent PoIs visited. Since the travel time overhead is known given a PoI visit order, the achieved Q_* measure of the new schedule can be computed by applying Theorem 4.6 with a modification for nonlinear periodic schedules.

For the case of the step utility function U_I , the QoM is in fact simply a *weighted* sum of the QoMs for the linear periodic subschedules which constitute the whole schedule (see the next theorem). For simplicity, we ignore the travel overhead (which can be easily incorporated). To set up the notation, for a general periodic schedule, let $\{p_k^i - q_k^i, q_k^i\}_{1 \leq k \leq K_i}$ be the consecutive sensor absent and present times for PoI i . Note that $P_* = \sum_{1 \leq k \leq K_i} p_k^i$ is the total period of the schedule (which is the same for all i 's). Then we have the following result.

THEOREM 5.2 (STEP UTILITY FUNCTION). *With the previous notation and the assumption of zero travel overhead, the QoM of PoI i is given by*

$$Q_i = \sum_{k=1}^{K_i} \frac{p_k^i}{P_*} \left[\frac{q_k^i}{p_k^i} + \frac{1}{p_k^i} \int_0^{p_k^i - q_k^i} \Pr(X \geq t) dt \right].$$

In particular, the QoM is a linear combination of the QoM of each individual sublinear periodic schedule which constitutes the overall nonlinear periodic schedule.

PROOF. The proof follows the same line as Theorem 4.2. The key observation that makes the proof go through is that if an event arriving during the absent period $p_k^i - q_k^i$ is ever captured, then it must be first captured in the next present period q_k^i . \square

5.3 Optimization of General Nonlinear Periodic Schedule: Simulated Annealing

We now illustrate how the above Theorem is used to optimize a general periodic schedule for Step utility. Starting with any initial periodic schedule of length l , there are $l!$ straightforward permutations of the schedule to obtain

```

Simulated Annealing Algorithm
1  best = s = initial periodic schedule
2  Qbest = Qs = QoM(best)
3  for (i = 0; i < computation_budget; i++)
4      p1, p2 = random positions in s
           subjected to selection criteria
5      new = s with p1, p2 swapped
6      if (new is physically infeasible)
7          continue
8      Qnew = QoM(new)
9      if (Qnew >= Qs)
10         s = new, Qs = Qnew
11         if (Qnew > Qbest)
12             best = new, Qbest = Qnew
13     else // simulated annealing
14         if (random < exp((Qnew - Qs) * i))
15             s = new, Qs = Qnew
16 return best

```

Fig. 11. Simulated annealing algorithm for optimal periodic schedule.

a general periodic schedule. An exhaustive search for an optimal schedule is computationally infeasible for large l . To overcome the challenge, we use a simulated annealing algorithm to search for a general nonlinear periodic schedule with its Q_* value as close to the optimal as possible. To apply this technique in our problem domain, care must be taken to consider the physical constraints of mobility including the finite speed of the sensor and the adjacencies of the PoIs.

The optimization algorithm is specified in Figure 11. We initialize the current search candidate s to some initial periodic schedule, and keep track of the current best schedule $best$ seen so far. We then randomly select two elements in s , say (L_i, C_i) and (L_j, C_j) , and swap $k_i\delta$ cover time from C_i with $k_j\delta$ time from C_j , to obtain a new schedule denoted by new , where $\delta = 2R/v_{\max}$, k_i and k_j are randomly selected positive integers such that $k_i\delta \leq C_i$ and $k_j\delta \leq C_j$. To avoid a cover time of less than δ for any element, we have the additional rule that any fractional δ time left by itself after a swap will be moved together with the associated whole number multiple of δ time moved. If two adjacent PoIs in new have distance ∞ between them, new is rejected as physically infeasible. Otherwise, we evaluate the Q_* of new . If new has a higher Q_* than s , we select new as shown. Otherwise, new is selected with a probability ($random$ in Line 13 is a random number in $[0,1]$). The search terminates after a given time budget.

For general utility functions, the closed analytical form of the QoM for a general (nonlinear) periodic schedule can be quite complicated. In particular, it will not be a weighted sum of the QoMs of the linear periodic subschedules. Nevertheless, one can still write down an analytical formula for the QoM by means of Theorem 5.3 and then resort to numerical integration to compute its value. The results can then be used as input for the simulated annealing algorithm.

THEOREM 5.3 (GENERAL UTILITY FUNCTION). *Let U^i be the utility function of the events at PoI i and $f(x)$ be the pdf of the event staying time. Then*

$$Q_i = \frac{1}{p_*} \int_0^{p_*} \int_0^\infty [U]^i(t, x) f(x) dx dt,$$

where $[U]^i(t, x) = U^i\left(\int_t^{t+x} p^i(s) ds\right)$ and $p^i(s)$ is a function that takes the value 1 when the sensor is present at PoI i at time s , and 0 otherwise.

PROOF. The proof is the same as Theorem 4.6 with the following understanding. The variable t refers to the event arrival time, x refers to the event staying time, and $\int_x^{x+t} p^i(s) ds$ is the total time the event is observed by the sensor. \square

Note that by increasing the period of the schedule for the optimization, the algorithm will optimize over an increasingly larger set of the plausible schedules. Hence, in principle, when the period is sufficiently large, the schedule obtained will give close to the optimal performance.

Remarks about Simulated Annealing. By now, simulated annealing and its extensions to stochastic combinatorial problems is a well-developed theory. Kirkpatrick et al. [1983] gives an early but still illustrative survey on the method. See also Cerny [1985] for an application to the Traveling Salesman Problem and Geman and Geman [1984] for a minimization problem in image processing. The main idea of this physically motivated approach is to employ the stochasticity of the algorithm to explore the whole state space so as not to get stuck in local minima. On the other hand, the “temperature” $T(t)$ of the algorithm is suitably reduced by some cooling schedule so that once the system gets close to the global minimum, the stochasticity is reduced and the state will stay close to the ultimate solution.

There are quite a few theoretical results for the convergence of the algorithm [Hajek 1988; Gidas 1985; Holley and Stroock 1988]. In these results, the temperature is cooled in accordance with a rate $\frac{C}{\log(t)}$ with the constant C chosen in accordance with the depths of the local minima in the energy landscape. However, the convergence statements are mostly of theoretical interest. Many practical logarithms often run faster (see the discussion of Brémaud [1999, pp 311–316]). Our simulated results clearly indicate so as recorded in Section 6.3.

Hence, we will refer to the body of theoretical work as a general benchmark but rely on the actual working simulations to provide practical guidance.

6. SIMULATION RESULTS

6.1 Single-PoI QoM

We present simulation results to illustrate the analytical results in Section 4. Recall the use of X and Y to denote the event staying and absent time variables, respectively. We measure the QoM Q_i achieved over 1,000,000 time units in a simulation run, and report the average Q_i of 10 different runs. The different

Table I. Maximum Available Information for Capture, Averaged over All the Simulated Events

Utility function	$X \sim Y \sim \text{Exp}(\lambda)$				$X \sim Y \sim \text{Pareto}(\alpha = 2, \beta)$			
	$\lambda = 0.25$	0.5	1	2	$\beta = 0.25$	0.5	1	2
Step	1	1	1	1	1	1	1	1
Exponential	0.95	0.91	0.83	0.72	0.84	0.97	1	1
Linear	0.89	0.79	0.63	0.43	0.44	0.75	1	1
S-shaped	0.88	0.78	0.62	0.40	0.36	0.82	1	1
Delayed Step	0.88	0.78	0.61	0.37	0.50	1	1	1

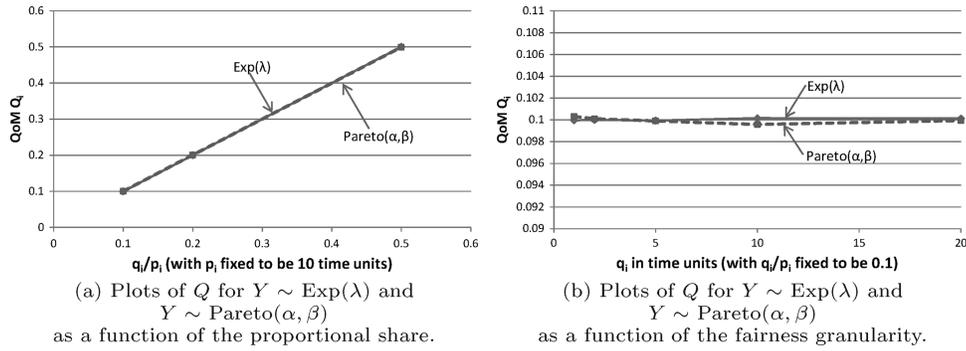


Fig. 12. Achieved QoM for blip events.

runs produce results that have extremely small differences. Hence, we omit the error bars in the reported results. Note that not all the events in a simulation stay long enough to be captured at the full utility. The maximum information available for capture is given by $\int_0^\infty U(x)f(x)dx$ as explained in Section 2.1, and the values obtained from the experiments are shown in Table I. Each reported experiment uses the same distribution for both the event staying and absent times, which is either Exponential with varying λ , or Pareto with varying β (and α is kept to be 2).

6.1.1 Blip Events. Figure 12 shows the QoM achieved for events that do not stay, for Exponential and Pareto event absent time distributions. For this type of events, full information about an event is captured instantaneously. Figure 12(a) shows that the QoM is directly proportional to the share q/p , and Figure 12(b) shows that the achieved QoM does not depend on the fairness granularity, as predicted by Theorem 4.1. The same results hold for different λ and (α, β) parameters of the Exponential and Pareto distributions, respectively.

6.1.2 Step Utility. We now present results for the step utility function U_I . Figures 13(a) and 13(b) show the achieved QoM as a function of the proportional share q/p for Exponential and Pareto event dynamics, respectively. The results agree with Theorem 4.2 and its instantiations for the distributions. Note that the fraction of events captured can be significantly higher than the proportional share, for example, a QoM of close to 0.4 is achieved for $\text{Exp}(\lambda = 0.25)$ and $\text{Pareto}(\alpha = 2, \beta = 2)$ even when the share is only slightly positive (see

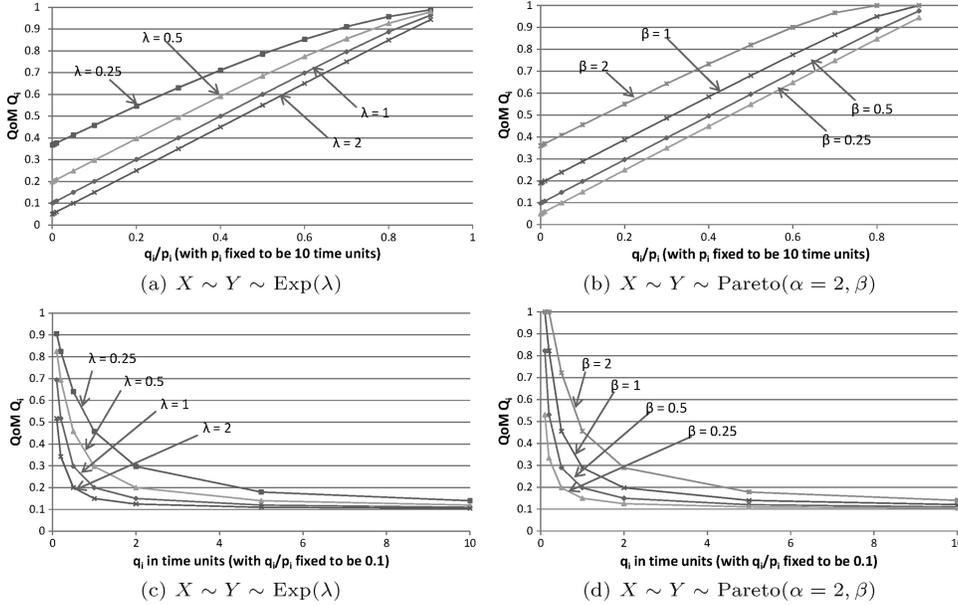


Fig. 13. Achieved QoM for events that stay and have the step utility function U_I .

Corollary 4.3). The observation time of the events increases as the events stay longer, and so the QoM is higher when λ is smaller for Exponential event dynamics and β is larger for Pareto event dynamics (see Section 4.3 (i)). In general, the QoM is not linear in the proportional share.

Figures 13(c) and 13(d) show the QoM as a function of the fairness granularity for Exponential and Pareto event dynamics, respectively. As predicted by Corollary 4.4, the QoM is a monotonically decreasing function of q (and hence, p , as we have q_i/p_i fixed), meaning that finer grained fairness will improve performance. As explained before, the QoM increases as λ decreases for the Exponential distribution and as β increases for the Pareto distribution. Furthermore, the QoM converges to the maximum value one and the proportional share $\gamma = q/p$ as q (and hence, p) converges to 0 and ∞ , respectively (see Corollary 4.4).

6.1.3 Exponential Utility. We now present results for the exponential utility function U_E (with $A = 5$). Figures 14(a) and 14(b) show the achieved QoM as a function of the proportional share for Exponential and Pareto event dynamics, respectively. Unlike Step utility, the achieved QoM is close to zero when the share is only slightly positive. This is due to the need to accumulate information for Exponential utility. As the share increases initially, however, there is a sharp gain in the QoM. This is because most information is gained during the initial observation of an event for Exponential utility. Moreover, the initial gain is higher when the events stay longer (i.e., smaller λ or larger β), because longer staying events are more likely to be captured even if they arrive when the sensor is not present. As the share further increases, the marginal gain in

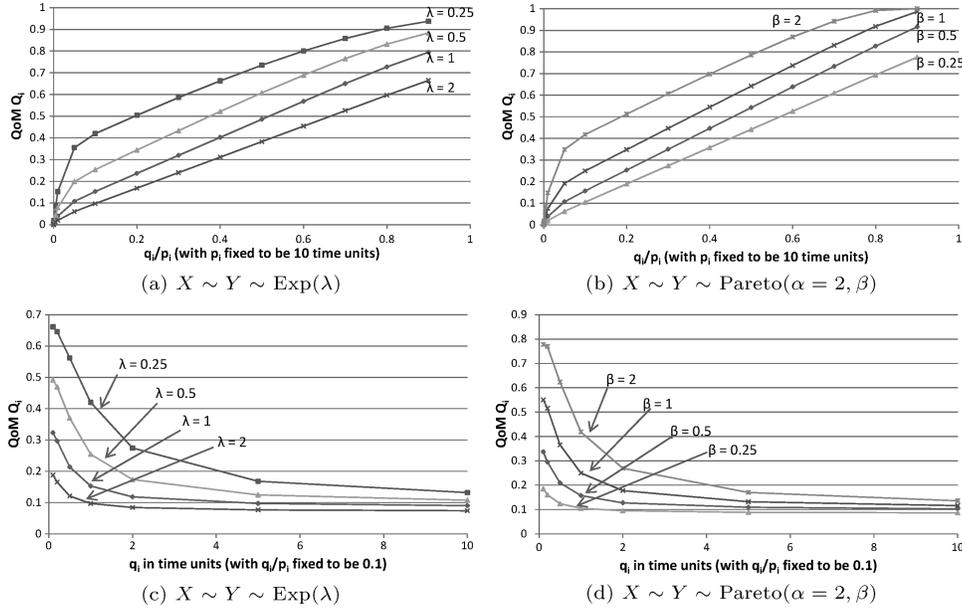


Fig. 14. Achieved QoM for events that stay and have the exponential utility function U_E , $A = 5$.

the QoM becomes smaller, again mimicking the decreasing marginal gain of information with longer observation time for the type of events. Note that for the larger λ values (e.g., $\lambda = 2$) or smaller β values (e.g., $\beta = 0.25$), the QoM is significantly smaller than one even for a large share. This is in part because at those parameter values, the events do not stay long enough to be captured at their full utility.

Figures 14(c) and 14(d) show the achieved QoM as a function of the fairness granularity for Exponential and Pareto event dynamics, respectively. For Exponential utility, the results agree with Eqs. (11) and (12). (See also Section 4.3(iv).) In particular, Figure 14(c) shows that the QoM is monotonically decreasing in q (and hence, p) and gives the correct QoM limits in Eq. (12) as $p \rightarrow 0$ and $p \rightarrow \infty$. In addition, the QoM increases when λ decreases. The results in Figure 14(d) show that similar results hold for Pareto event dynamics.

6.1.4 Linear Utility. We now present results for the linear utility function U_L (with $M = 1$). Figures 15(a) and 15(b) show the achieved QoM as a function of the proportional share for Exponential and Pareto event dynamics, respectively. Figures 15(c) and 15(d) show the achieved QoM as a function of the fairness granularity for the two types of event dynamics. The results are similar to Exponential utility. In particular, the QoM is a monotonically decreasing function of p , although it is flat over an initial range of p values, showing that there is no need for the sensor to move faster and achieve a smaller p after some point.

6.1.5 Delayed Step Utility. We now present simulation results for the delayed step utility function U_D ($D = 0.5$ time units). Figures 16(a) and 16(b)

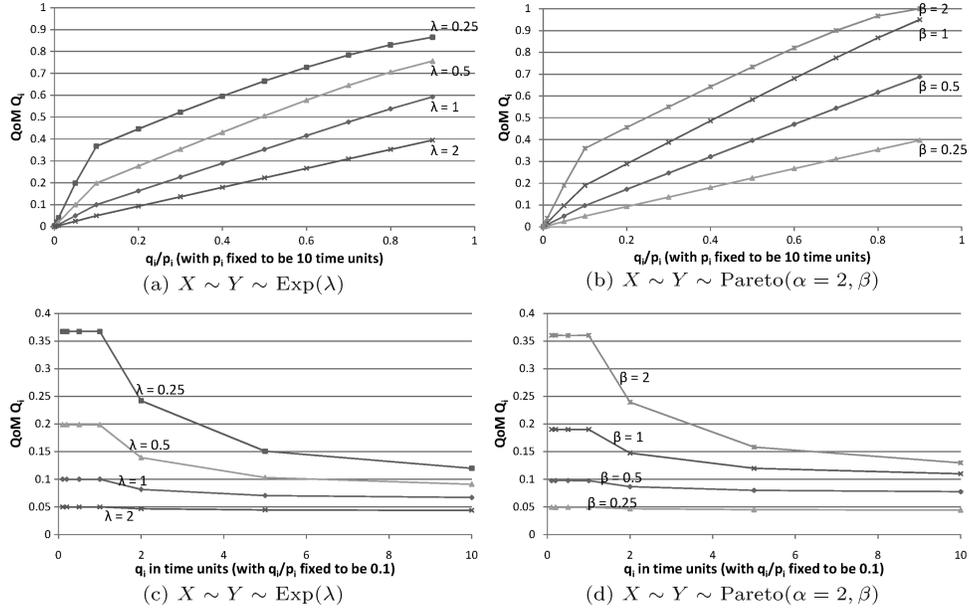


Fig. 15. Achieved QoM for events that stay and have the linear utility function U_L , $M = 1$.

show the achieved QoM as a function of the proportional share for Exponential and Pareto event dynamics, respectively. They show that the QoM is monotonically increasing in the proportional share, and the QoM is higher when the events stay longer (i.e., smaller λ or larger β).

Figures 16(c) and 16(d) show the achieved QoM as a function of the fairness granularity. Note that in this case, the QoM is no longer monotonically decreasing in p , but the optimal fairness occurs at an intermediate value. Note also that for $\lambda = 2 = \frac{1}{D}$, the QoM is a constant function of p for large p . These properties are all discussed in Section 4.3(v).

6.1.6 S-Shaped Utility. Figure 17 presents the QoM results for the S-shaped utility function U_S . Although we do not have corresponding analytical results for S-shaped utility, note from Figure 17 that the results are similar to Delayed Step utility. This is due to the resemblance between the two utility functions.

6.2 Multiple Sensor Scenario

We now evaluate the multiple-sensor scenario discussed and analyzed in Section 4.4. We first summarize the simulation results and their interpretation, before presentation of the detailed results.

- (1) The QoM is a monotonically increasing function of m . This is easily understood as the effective coverage time is equal to mq which is proportional to the number of sensors present.
- (2) For *concave* utility functions, such as the step, exponential, and linear functions, the QoM decreases with p and r . This is because a larger p effectively

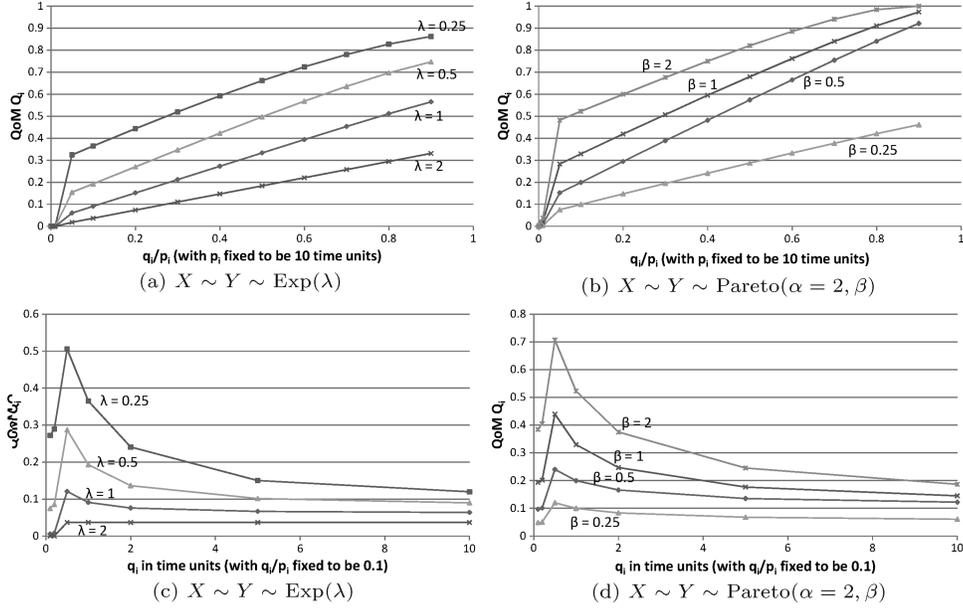


Fig. 16. Achieved QoM for events that stay and have the delayed step utility function U_D , $D = 0.5$.

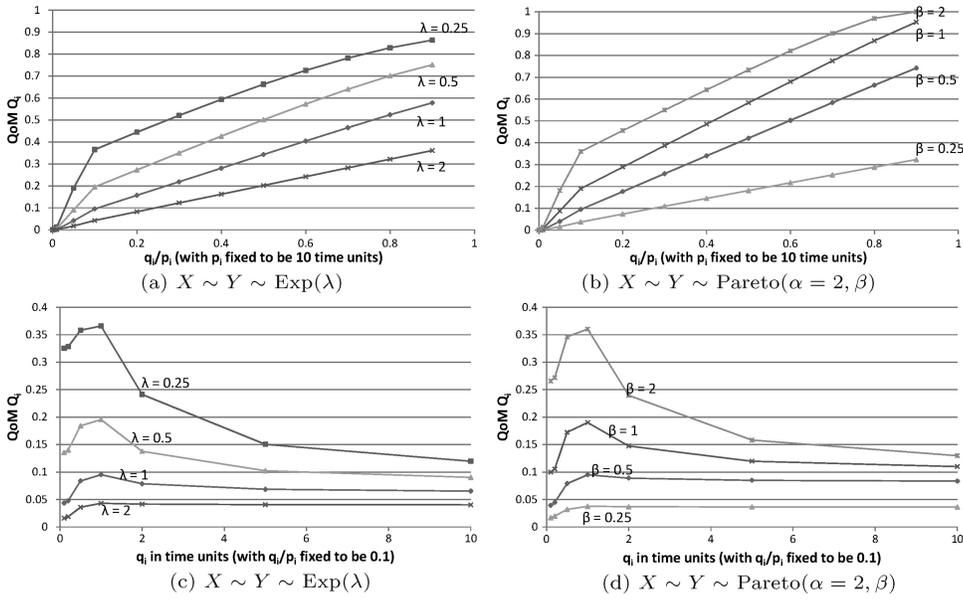


Fig. 17. Achieved QoM for events that stay and have the S-shaped utility function U_S .

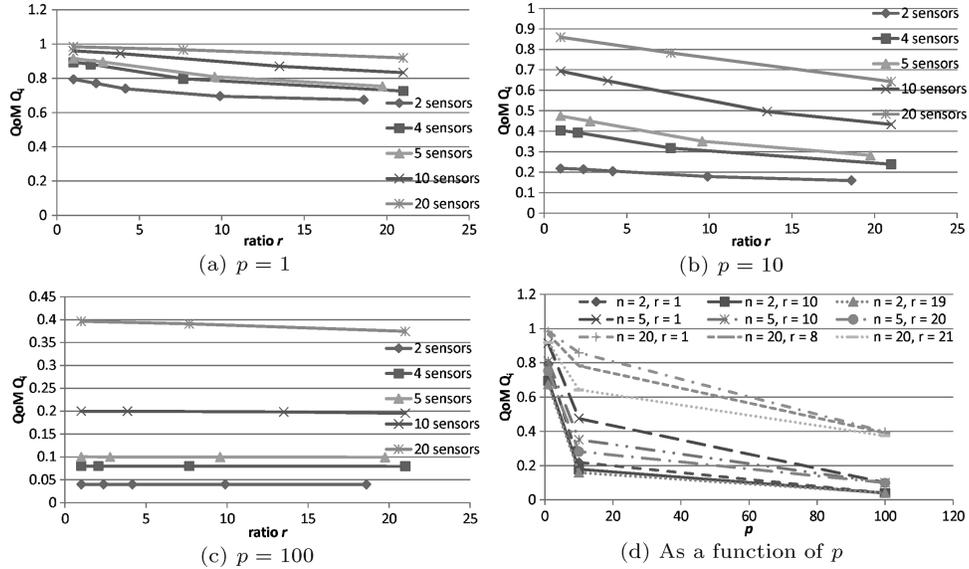


Fig. 18. Achieved deployment QoM Q_i for staying events with step utility function U_I . $q/p = 0.01$, $X \sim Y \sim \text{Exp}(\lambda = 1)$.

leads to a coarser fairness granularity. By the analysis in Section 4.4, for concave utility functions, a finer granularity gives a higher QoM. Similarly, a larger r means that the coverage times are more clustered together, that is, $a \ll b$. This is equivalent to a coarser granularity leading to missed events and also redundancy in observing the same events.

- (3) For nonconcave utility functions such as the delayed step and S-shaped ones, the QoM initially increases and then decreases with r . The explanation is similar to the analysis in Section 4.4. In order to gain a significant amount of information about a single event, enough observation time must be achieved. This requires larger values of r . After the critical time is passed, a higher degree of clustering leads to redundancy in collecting the information.

We now proceed to discuss the detailed results. These results are all supported by the analysis in Section 4.4. As mentioned there, we will emphasize the behavior of the QoM as a function of the parameters p , m , and r . We consider the cases when there are 2, 4, 5, 10, 20, and 40 mobile sensors. We measure the QoM Q_i achieved over 1,000,000 time units in a simulation run, and report the average Q_i of 20 different runs. The event staying and absent times are both Exponentially distributed with $\lambda = 1$, and the share $q/p = 0.01$.

6.2.1 Nondecreasing Concave Utility Functions. We present results for the nondecreasing concave utility functions, namely, the step utility function U_I , exponential utility function U_E (with $A = 5$), and linear utility function U_L (with $M = 1$). Figures 18, 19, and 20 show the achieved QoM as a function

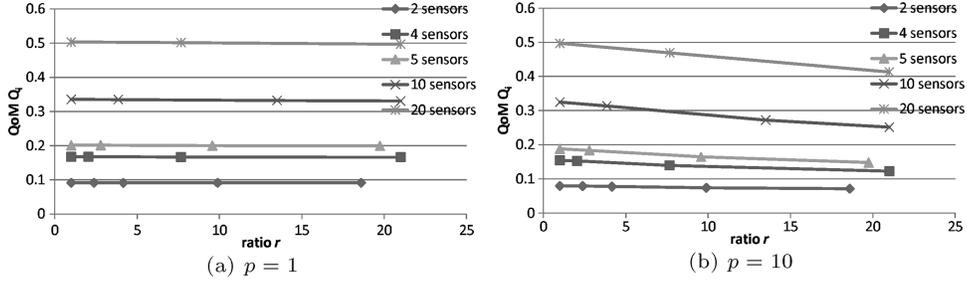


Fig. 19. Achieved deployment QoM Q_i for staying events with exponential utility function U_E , $A = 5$, $q/p = 0.01$, $X \sim Y \sim \text{Exp}(\lambda = 1)$.

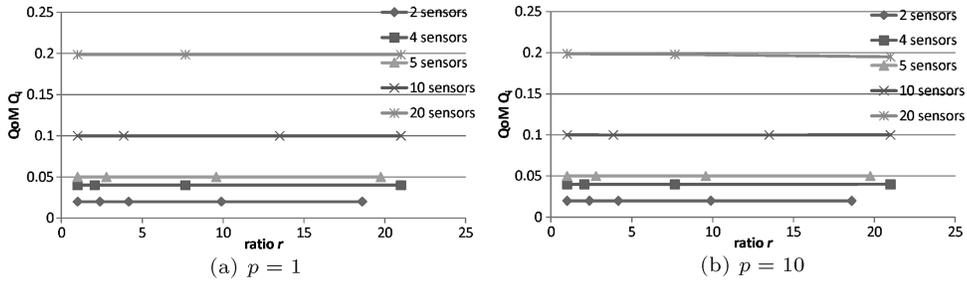


Fig. 20. Achieved deployment QoM Q_i for staying events with linear utility function U_L , $M = 1$, $q/p = 0.01$, $X \sim Y \sim \text{Exp}(\lambda = 1)$.

of the number of mobile sensors m and the clustering ratio r for the Step, Exponential, and Linear utility functions, respectively. The figures show that the achieved deployment QoM Q_i is the highest when the visits are evenly distributed, that is, $r = 1$. As the visits are more clustered together, that is, r is larger, Q_i is worsened when the interval of consecutive visits, if they are evenly distributed, is comparable to the event staying time. It is because when we have a concave utility function, it is generally more productive to monitor more events than to observe the same event for longer. For Linear utility, however, notice from Figure 20 that the separation of consecutive visits to a PoI has insignificant effects on the achieved deployment QoM. This is due to the nature of the linear utility function, that is, it is indifferent to Q_i whether the same event or different events are monitored.

The figure also shows that the achieved deployment QoM Q_i may grow sub-linearly with the number of sensors. It is because as the interval of consecutive visits, if they are evenly distributed in one period, is comparable or smaller than the event staying time, then it is more likely for the sensors to capture the same event, which results in no improvement in Q_i .

When the visit time of a sensor at the PoI is comparable to the event dynamics as depicted in Figure 18(c), Q_i can only be slightly worsened, if it is affected at all, if the sensors are more clustered together. It is because in such a situation, different visits to the PoI are likely to observe different events, and a more clustered group of sensors only reduces the probability of capturing new events.

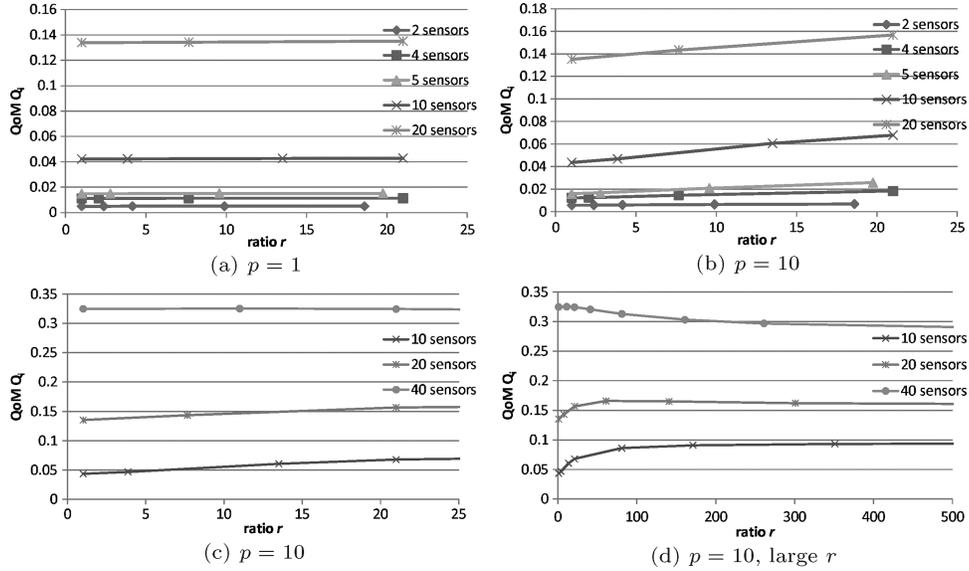


Fig. 21. Achieved deployment QoM Q_i for staying events with S-shaped utility function U_S . $q/p = 0.01$, $X \sim Y \sim \text{Exp}(\lambda = 1)$.

6.2.2 Utility Functions as a Combination of Convex and Then Concave Parts. We now present results for the S-shaped utility function U_S and the delayed step utility function U_D ($D = 0.5$ time units). Each can be seen as a combination of two different parts, a convex part followed by a concave part. Figures 21 and 22 depict the achieved QoM as a function of the number of mobile sensors m and the clustering ratio r for the S-shaped and delayed step functions, respectively. The figures show that clustering sensors together can improve the achieved deployment QoM as the separation of consecutive visits, when they are evenly distributed, is comparable to the event present time. It is because in such a scenario, clustering sensors together can help monitor the same event for longer in the convex part of the utility function, so that it is more likely to approach the information threshold for a better Q_i . However, Figures 21(d) and 22(d) show that when the sensor visits are too clustered together, so that they collect information in the concave part of the utility function instead of capturing new events in the convex part of the function, the overall Q_i will suffer.

Figure 22 also shows that the achieved deployment QoM Q_i may grow super-linearly with the number of sensors. It is because as the interval of consecutive visits, when they are evenly distributed in one period, is comparable or smaller than the event staying time, then it is more likely for the sensors to monitor the same event and aggregate information within the threshold for productive information gain, which results in a sharp increase in the utility and an improvement in Q_i .

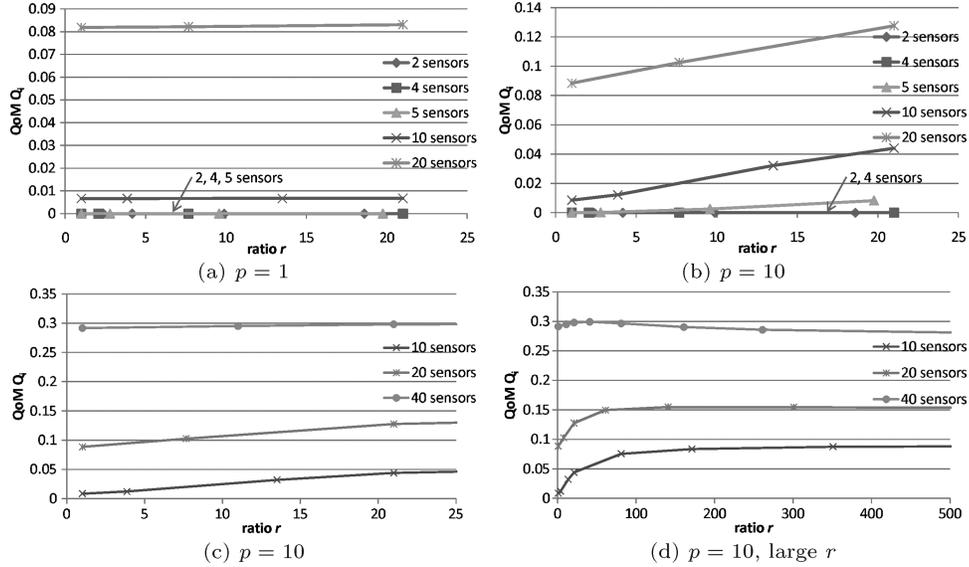


Fig. 22. Achieved deployment QoM Q_i for staying events with delayed-step utility function U_D , $D = 0.5$ time units. $q/p = 0.01$, $X \sim Y \sim \text{Exp}(\lambda = 1)$.

6.3 General Nonlinear Periodic Coverage Optimization: Simulated Annealing

We present two sets of simulation results to illustrate the performance of the optimization algorithm in Section 5 for periodic schedules. In the first set of simulations, we use n PoIs, denoted as $1, 2, \dots, n$, such that $d_{ij} = 2R + O$ for $i \neq j$, where R is the sensing range, and O is the travel overhead (i.e., the distance in E_{ij} where the sensor is not covering any PoI). We assume that the maximum speed of the sensor is such that it takes the sensor one time unit to cover a distance of $2R$, so that the minimum staying time of the sensor at any PoI in a coverage schedule is $\delta = 1$ time unit. In the second set of simulations, we use real city maps of dimensions 2000 m by 2000 m divided into cells of dimensions 250 m by 250 m. Each cell is a PoI, and its threat level is set to be the estimated population size in that cell. The distance d_{ij} is measured between the center of cell i and that of cell j . We compare the achieved QoM by varying the sensor speed and the minimum staying time of the sensor at the PoIs. Note that we consider large-scale city maps with highly unstructured placements of the PoIs because such a setting is most challenging for our problem. It is likely that our solution can be more easily applied in other smaller and/or more structured, but still realistic, deployments, for example, the monitoring of a given set of rooms in the same building. For each experiment, we report the average of 20 runs of the algorithm. The differences between the measurements are small. We will thus omit the error bars, although in the case of the deployment QoM, we will also report the maximum Q_* achieved in the 20 runs. Results of the first set of simulations are discussed in Sections 6.3.1, 6.3.2, and 6.3.3. Those of the second set of simulations are discussed in Section 6.3.4.

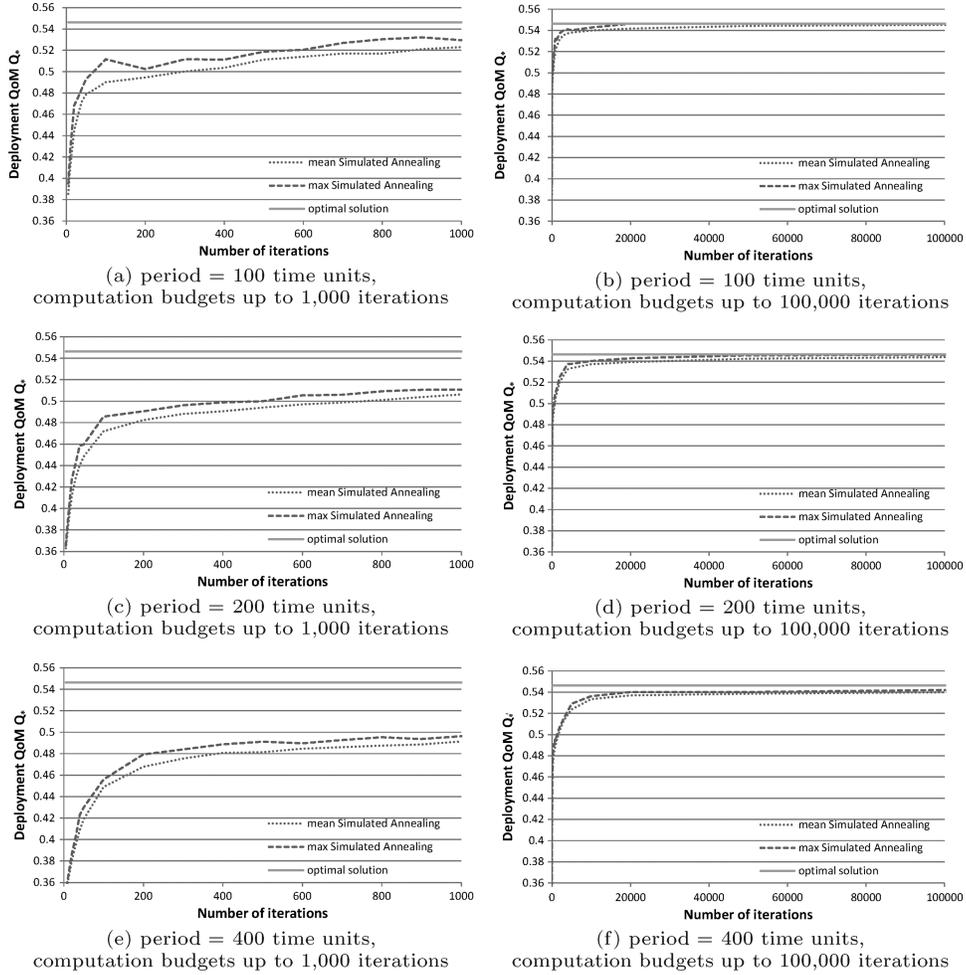


Fig. 23. Achieved deployment QoM Q_* for staying events with Step utility and proportional share ratios of 50:49:1. $X \sim Y \sim \text{Exp}(\lambda = 1)$.

6.3.1 Revisit of Example (Section 5.2). This example motivates the use of optimized general periodic schedules. We have three PoIs, and the proportional shares of 1, 2, and 3 are in ratios of 50:49:1. We do not consider travel overhead in this set of experiments, that is, $O = 0$. We show the optimizations over schedules of period l , where $l = 100, 200$, and 400 time units. The algorithm in Figure 11 is run with the initial schedule set to be the optimal *linear* periodic schedule of the given length. Figures 23(a), 23(c), and 23(e) plot the maximum and average deployment QoM Q_* achieved by the simulated annealing algorithm for small computation budgets of up to 1000 iterations. The optimal deployment QoM is also shown as the horizontal green line in the figure. Figures 23(b), 23(d), and 23(f) plot the corresponding results for larger computation budgets of up to 100,000 iterations.

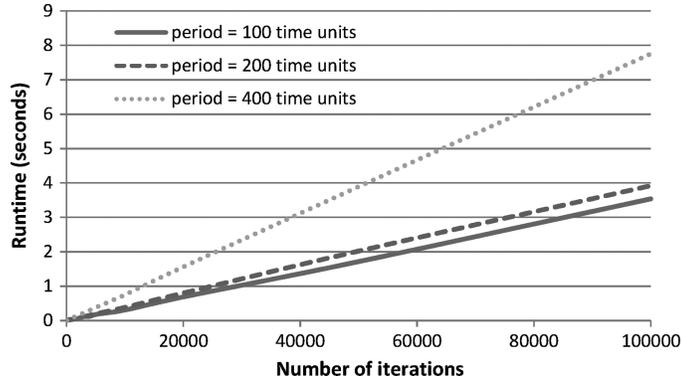


Fig. 24. Execution time of simulated annealing algorithm for staying events with Step utility and proportional share ratios of 50:49:1. $X \sim Y \sim \text{Exp}(\lambda = 1)$.

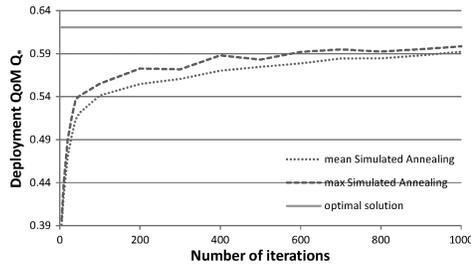
From the smaller computation budget results, note that the optimal linear periodic schedule is suboptimal in general but the simulated annealing can produce schedules that rather quickly approach the optimal as the number of iterations increases. From the larger computation budget results, note that when the number of iterations is large enough, the simulated annealing can find a solution extremely close to the optimal (within 2%). When $l = 400$ time units, the results are similar and a close-to-optimal solution is found within 100,000 iterations. Initially, however, Q_* increases more slowly with the number of iterations than $l = 100$ time units. This is because in this particular experiment, the globally optimal schedule can be found with a period length of 100 time units. Increasing the optimization period to 400 time units will not increase the potential to find a better solution, but will increase the search space for the optimal solution.

We have measured the run time of the simulated annealing, written in C#, on a Pentium-4 3.4-GHz PC with L1/L2 cache sizes of 8 KB/512 KB and 2 GB of RAM. The results, shown in Figure 24, indicate that the run time is linear in the number of iterations, and is about 3.5 s and 7.7 s for 100,000 iterations and an optimization period of 100 and 400 time units, respectively. Figure 24 shows the execution time of the simulated annealing algorithm for different lengths of the periodic schedule being optimized. From the figure, note that the execution time is roughly linear in the number of iterations. Also, the proportionality constant increases when the length of the periodic schedule increases.

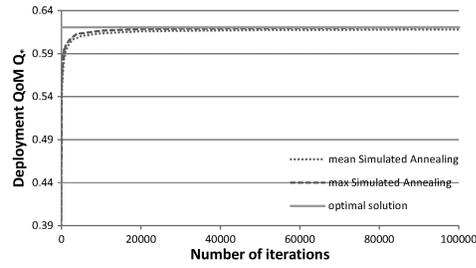
6.3.2 Other Numbers of PoIs and Proportional Share Ratios (No Travel Overhead). We vary the number of PoIs and their proportional share ratios in a number of experimental schedules listed in Table II, when there is no travel overhead (i.e., $O = 0$). The results for 3, 5, and 10 uniform PoIs (i.e., the PoIs get equal proportional shares, corresponding to experimental schedules I, II, and IV) are shown in Figure 25. Small computation budget results are shown on the left column; corresponding large computation budget results are

Table II. Experimental Schedules

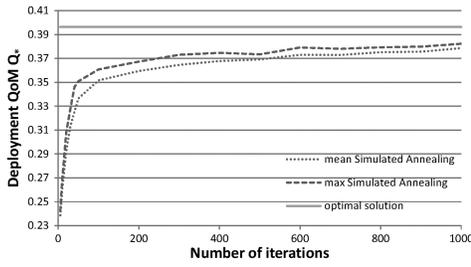
Schedule ID	# PoIs	Proportional share ratios
I	3	0.34:0.33:0.33
II	5	0.2:0.2:0.2:0.2:0.2
III-a	5	0.36:0.16:0.16:0.16:0.16
III-b	5	0.52:0.12:0.12:0.12:0.12
III-c	5	0.68:0.08:0.08:0.08:0.08
III-d	5	0.84:0.04:0.04:0.04:0.04
IV	10	0.1:0.1:0.1:0.1:0.1:0.1:0.1:0.1:0.1:0.1



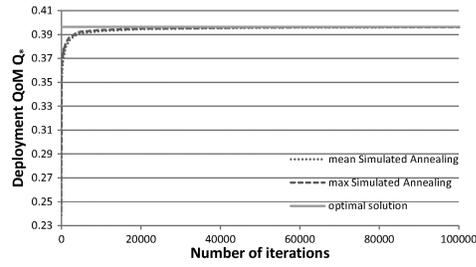
(a) Schedule I: 3 uniform PoIs, computation budgets up to 1,000 iterations



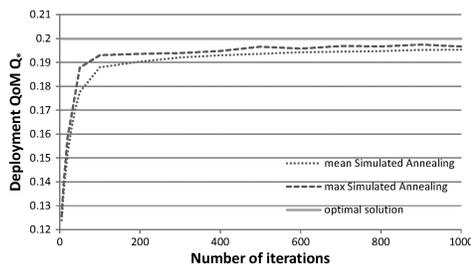
(b) Schedule I, computation budgets up to 100,000 iterations



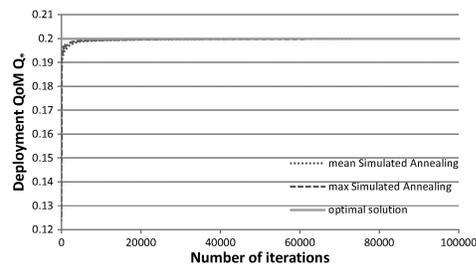
(c) Schedule II: 5 uniform PoIs, computation budgets up to 1,000 iterations



(d) Schedule II, computation budgets up to 100,000 iterations



(e) Schedule IV: 10 uniform PoIs, computation budgets up to 1,000 iterations



(f) Schedule IV, computation budgets up to 100,000 iterations

Fig. 25. Achieved deployment $QoM Q_*$ for staying events with Step utility and different numbers of PoIs of equal proportional shares. $X \sim Y \sim \text{Exp}(\lambda = 1)$, period = 100 time units.

shown on the right column. Figure 26 shows the small computation budget results for 5 PoIs, with one of the PoIs getting a larger proportional share and the remaining 4 getting the same smaller proportional share. The fraction of coverage time given to the higher-share PoI increases progressively from

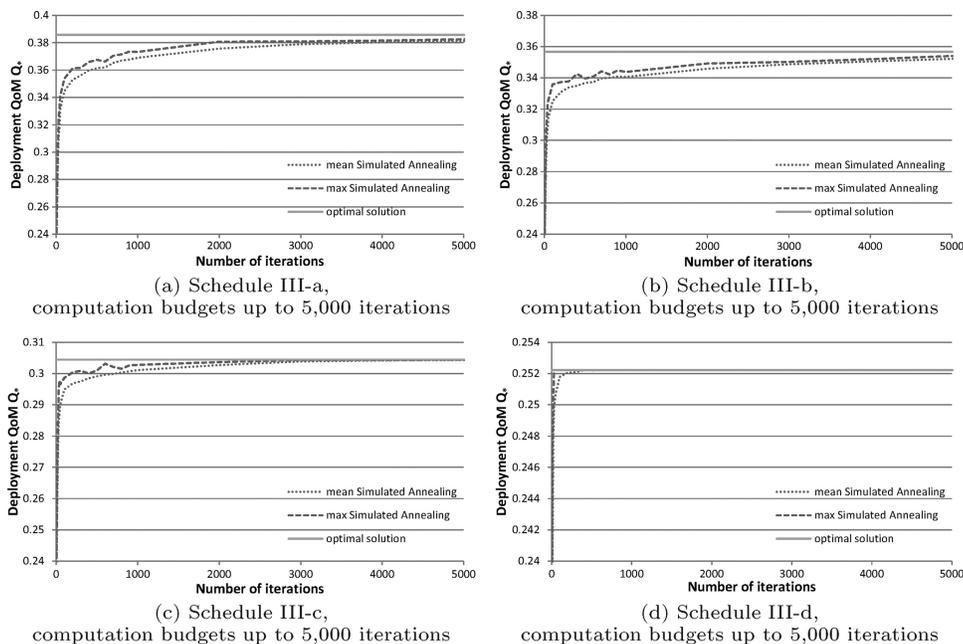


Fig. 26. Achieved deployment QoM Q_* for staying events with Step utility. There are 1 higher-share PoI and 4 lower-share PoIs, and the coverage-time bias towards the higher-share PoI increases from (a) to (d). $X \sim Y \sim \text{Exp}(\lambda = 1)$, period = 100 time units.

Figure 26(a) to Figure 26(d), corresponding to the experimental schedules III-a–III-d in Table II. The results show that the optimization algorithm approaches the optimal Q_* quickly and a solution extremely close to the optimal is found within a few thousand iterations. Figure 25 shows that when the number of PoIs covered increases, the QoM drops generally, because each PoI is visited less frequently. The figure shows also that the optimal schedule is reached earlier when there are more PoIs. It is because when the number of PoIs increases, the chance to break up a long and continuous stay at a PoI into a number of shorter visits also increases, which results in finer grained sharing of coverage time between the PoIs and therefore higher QoM. Figure 26 shows that when the fraction of coverage time is more heavily biased towards the higher-share PoI, the optimal solution can be obtained in a smaller number of iterations. It is because when the bias increases, the optimal schedule in which visits to the lower-share PoIs are the shortest possible can be discovered more easily. Note that the total QoM drops as the bias increases. It is because all the lower-share PoIs are visited less frequently although they present better opportunities for information capture compared with the higher-share PoI.

Figure 27 shows the execution time of the simulated annealing algorithm for different numbers of PoIs. Notice that as the number of PoI increases, the running time of the algorithm increases.

6.3.3 Impact of Travel Overhead. We now study the effects of the travel overhead O using the experimental schedule IV in Table II. We vary O to be

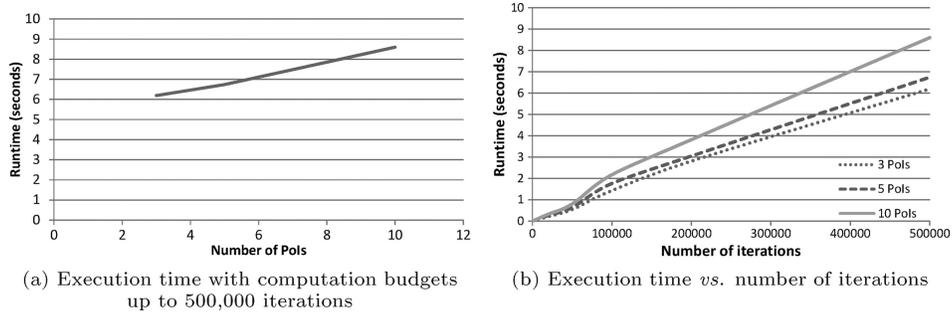


Fig. 27. Execution time of simulated annealing algorithm for staying events with Step utility and equal proportional shares of coverage time between the PoIs. $X \sim Y \sim \text{Exp}(\lambda = 1)$, period = 100 time units.

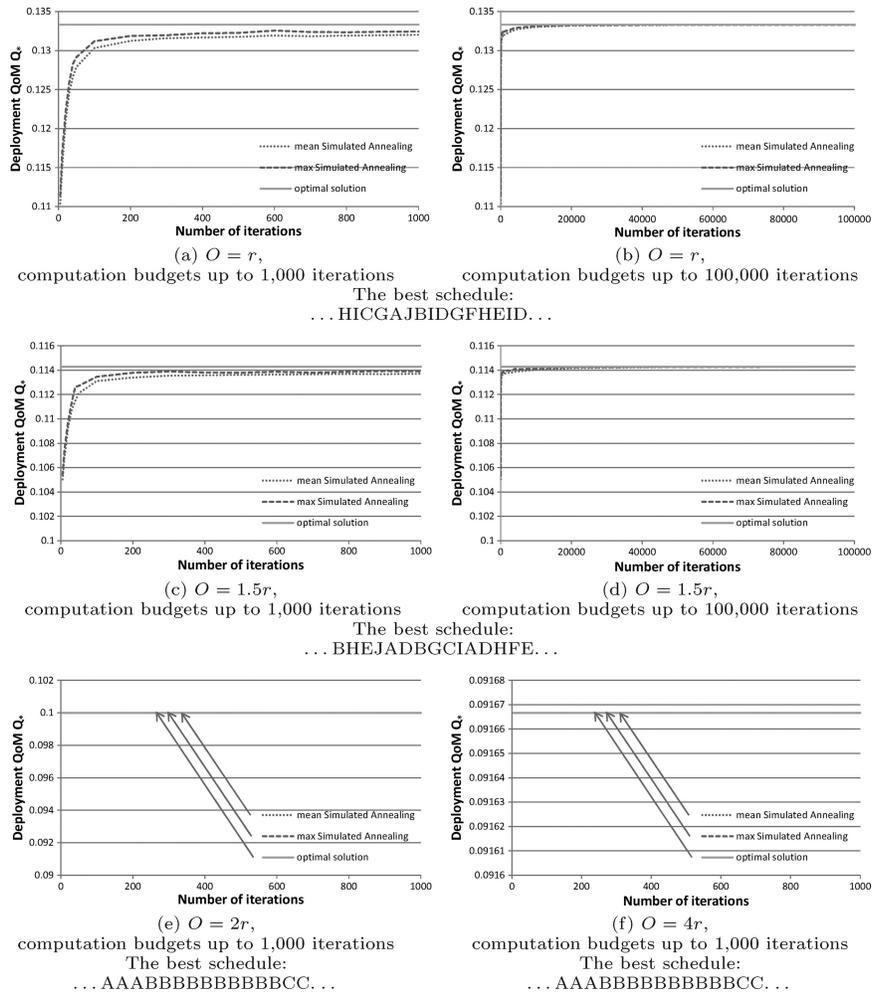


Fig. 28. Achieved deployment QoM Q_* for staying events with Step utility and experimental schedule IV, as the travel overhead O varies. $X \sim Y \sim \text{Exp}(\lambda = 1)$, period = 100 time units.

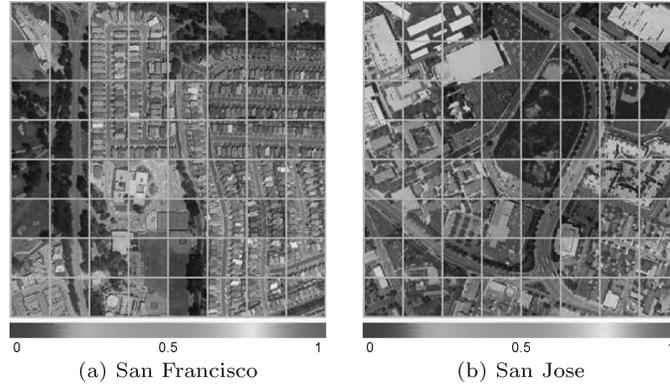
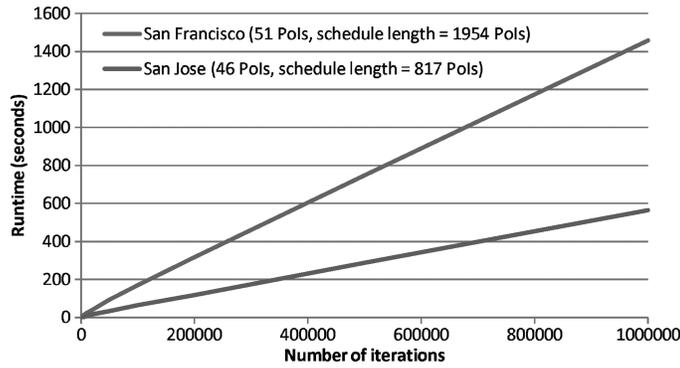


Fig. 29. City maps with the associated threat profile.

Fig. 30. Execution time of simulated annealing algorithm for staying events with Step utility using real city maps. $X \sim \text{Exp}(\lambda = 1 \text{ hour})$.

R , $1.5R$, $2R$, and $4R$. The results are shown in Figure 28. From the figure, note that when the overhead is $2R$ or larger, the optimal linear periodic schedule (recall that we initialize the simulated annealing with the optimal linear periodic schedule in each experiment) is globally optimal. It is because when the travel overhead is high, the cost of moving frequently between different PoIs outweighs the potential gain in QoM due to finer grained sharing between the PoIs. Otherwise (i.e., $O < 2R$), the optimal schedule is one in which there is maximum interleaving of visits to the different PoIs. This is because the interleaving leads to shorter but more frequent visits to the same PoIs, and is beneficial for information capture of staying events for the step utility.

6.3.4 Real City Maps. We now study the performance of the algorithm using the real city maps shown in Figure 29. The color of a cell in the maps represents the estimated relative population residing in that cell, with a reddish color denoting a larger population. We vary v to be 8 km/h and 16 km/h, δ to be 1 min, 2 mins, 5 mins, and 15 mins, and the event staying time is Exponentially distributed with the mean value α equal to one hour. The average runtime of the algorithm is given in Figure 30, and the achieved QoMs are shown in Figures 31 and 32 for San Francisco and San Jose, respectively. Figure 30 shows

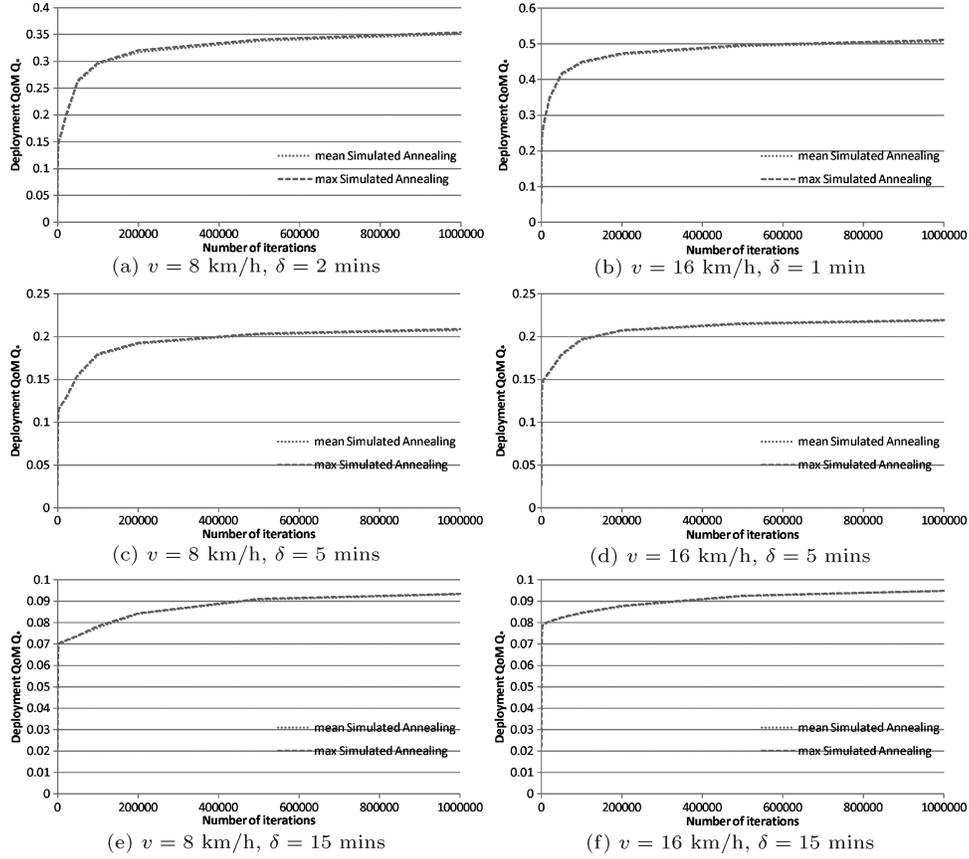


Fig. 31. Achieved deployment QoM Q_* for staying events with Step utility using the city map of San Francisco. $X \sim \text{Exp}(\lambda = 1 \text{ hour})$.

that a longer schedule length takes a longer time for the search algorithm to run, and the runtime roughly grows linearly with the length. From Figures 31 and 32, we can observe that the algorithm is able to compute a much better schedule than the linear periodic one using a relatively short time as depicted in Figure 30. Figures 31 and 32 show that a faster speed improves the deployment QoM Q_* as the period length is reduced. A better deployment QoM Q_* can also be achieved more significantly by a shorter staying time at a PoI.

7. CONCLUSIONS

We have presented extensive analysis to understand the QoM properties of proportional-share mobile sensor coverage. We show that: (1) A higher share of the coverage time generally increases the QoM, but the relationship is not linear except for blip events; (2) For staying events, the QoM can be much higher than the proportional share, due to the observation of “extra” events that arrive when the sensor is not present. This justifies mobile coverage from an information-capture point of view, that is, the sensor gains by moving between places to search for new information; (3) The event utility function is

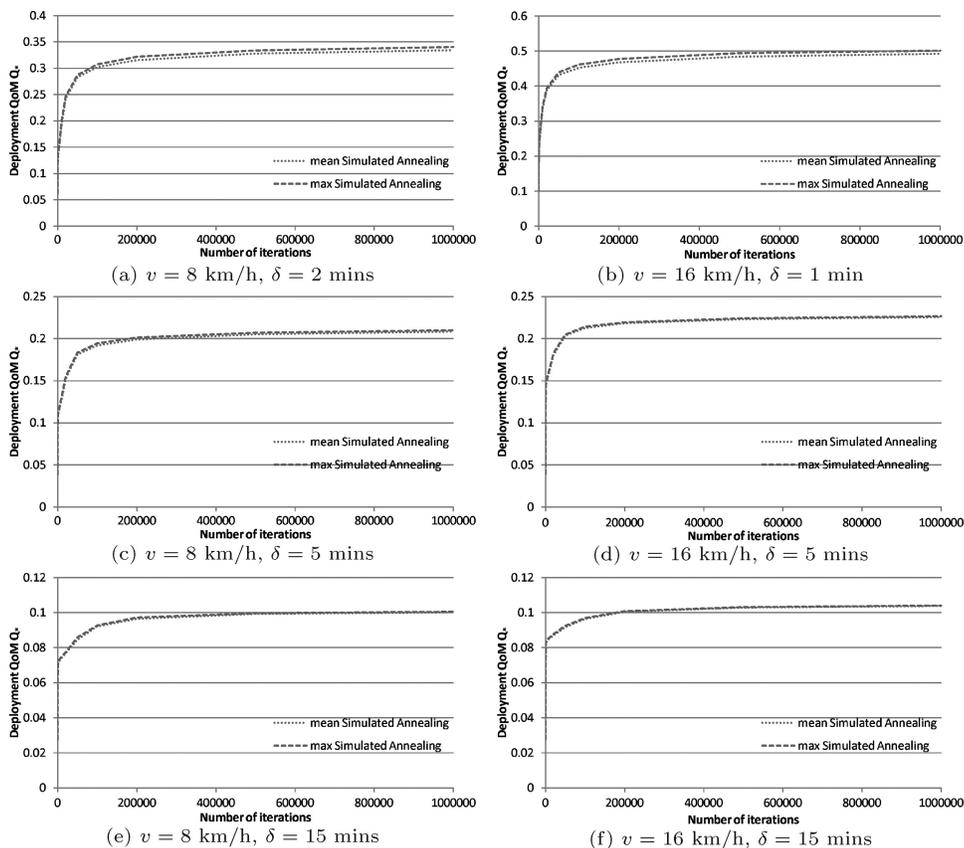


Fig. 32. Achieved deployment QoM Q_* for staying events with Step utility using the city map of San Jose. $X \sim \text{Exp}(\lambda = 1 \text{ hour})$.

important in determining the optimal fairness granularity p . For concave utility functions such as Step, Exponential, and Linear utilities, the QoM monotonically decreases with p , whereas for Delayed Step and S-Shaped utilities, the QoM generally peaks at an intermediate p . Our analysis for Exponential/Pareto event dynamics and different forms of the utility function is all supported by the simulation results. We presented optimization algorithms for both linear and general proportional-share periodic coverage. Implementation results show that the simulated annealing algorithm can efficiently compute a periodic schedule that practically maximizes the total QoM, even for huge search spaces implied by long scheduling periods. We also illustrated the performance and efficiency of the simulated annealing algorithm as a function of the number of PoIs, the travel overhead, and the distribution of proportional shares among the PoIs.

REFERENCES

BATALIN, M. A., AND SUKHATME, G. S. 2003. Multi-robot dynamic coverage of a planar bounded environment. Tech. rep. CRES-03-011, USC Viterbi.

- BISNIK, N., ABOUZEID, A., AND ISLER, V. 2006. Stochastic event capture using mobile sensors subject to a quality metric. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*. ACM, New York.
- BRÉMAUD, P. 1999. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, Berlin.
- CARLSSON, S., NILSSON, B. J., AND NTAFOIS, S. C. 1991. Optimum guard covers and mwatchmen routes for restricted polygons. In *Proceedings of the Workshop on Algorithms and Data Structures*.
- CARLSSON, S., JONSSON, H., AND NILSSON, B. J. 1993. Finding the shortest watchman route in a simple polygon. In *Proceedings of the International Symposium on Algorithms and Computation (ISAAC)*.
- CERNY, V. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J. Optimiz. Theory Appl.* 45, 1.
- CHAKRABORTY, S., DONG, Y., YAU, D. K. Y., AND LUI, J. C. S. 2006. On the effectiveness of movement prediction to reduce energy consumption in wireless communication. *IEEE Trans. Mobile Comput.* 5, 2.
- CHENG, W., LI, M., LIU, K., LIU, Y., LI, X., AND LIAO, X. 2008. Sweep coverage with mobile sensors. In *Proceedings of the International Parallel and Distributed Processing Symposium (IEEE IPDPS)*. IEEE Computer Society Press, Los Alamitos, CA.
- COCHRAN, E. 1960. New concepts of the learning curve. *J. Indust. Eng.* 11, 4, 317–327.
- COSTA, P., FREY, D., MIGLIAVACCA, M., AND MOTTOLA, L. 2006. Towards lightweight information dissemination in inter-vehicular networks. In *Proceedings of the ACM International Workshop on Vehicular Internetworking (VANET)*.
- CROVELLA, M. E., AND BESTAVROS, A. 1996. Self-similarity in world wide web traffic evidence and possible causes. *Trans Netw.*
- DAI, F., WU, J., AND YANG, S. 2007. Logarithmic store-carry-forward routing in mobile ad hoc networks. *IEEE Trans. Parall. Distrib. Syst.* 18, 6.
- EFRAT, A., GUIBAS, L. J., HAR-PELED, S., LIN, D. C., MITCHELL, J. S. B., AND MURALI, T. M. 2000. Sweeping simple polygons with a chain of guards. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*. ACM, New York.
- GEMAN, S., AND GEMAN, D. 1984. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 6, 721–741.
- GIDAS, B. 1985. Nonstationary markov chains and convergence of simulated annealing algorithm. *J. Stat. Phys.* 39, 1/2, 73–131.
- GRIBBLE, S. D., MANKU, G. S., ROSELLI, D., BREWER, E., GIBSON, T. J., AND MILLER, E. L. 1998. Self-similarity in file systems. In *Proceedings of the ACM SIGMETRICS*. ACM, New York.
- GUIBAS, L., LATOMBE, J., LAVALLE, S., LIN, D., AND MOTWANI, R. 1999. A visibility-based pursuit-evasion problem. *Int. J. Computat. Geom. Appl.*
- GUPTA, H., DAS, S., AND GU, Q. 2003. Connected sensor cover: self organization of wireless sensor networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.
- GUPTA, V., JEFFCOAT, D., AND MURRAY, R. 2006. On sensor coverage by mobile sensors. In *Proceedings of the IEEE Conference on Decision and Control*. IEEE Computer Society Press, Los Alamitos, CA.
- HAJEK, B. 1988. Cooling schedules for optimal annealing. *Math. Oper. Res.* 13, 2, 311–329.
- HE, G., AND HOU, J. C. 2005. Tracking targets with quality in wireless sensor networks. In *Proceedings of IEEE Conference on Network Protocols (ICNP)*. IEEE Computer Society Press, Los Alamitos, CA.
- HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. 2003. Range-free localization schemes for large scale sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, New York.
- HOLLEY, R., AND STROOCK, D. 1988. Simulated annealing via sobolev inequalities. *Comm. Math. Phys.* 115, 553–569.
- HOWARD, A., MATARIC, M., AND SUKHATME, G. 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS)*.

- HULL, B., BYCHKOVSKY, V., ZHANG, Y., CHEN, K., GORACZKO, M., MIU, A. K., SHIH, E., BALAKRISHNAN, H., AND MADDEN, S. 2006. Cartel: A distributed mobile sensor computing system. In *Proceedings of ACM SenSys*. ACM, New York.
- JEFFAY, K., SMITH, F. D., MOORTHY, A., AND ANDERSON, J. 1998. Proportional share scheduling of operating system services for real-time applications. In *Proceedings of the IEEE Real-Time System Symposium (RTSS)*. IEEE Computer Society Press, Los Alamitos, CA.
- KIRKPATRICK, S., GELATT, C. D., AND VECCHI, M. P. 1983. Optimization by simulated annealing. *Science, New Series* 220, 4598, 671–680.
- PAREKH, A. K., AND GALLAGER, R. G. 1993. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Trans Network*. 1, 3.
- LAPP, R. E., AND ANDREWS, H. L. 1948. *Nuclear Radiation Physics*. Prentice-Hall, Englewood, Cliffs, NJ.
- LEE, R. W., AND KULESZ, J. J. 2006. A risk-based sensor placement methodology. Tech. rep., Computational Sciences and Engineering Division, ORNL.
- LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. 1993. On the self-similar nature of ethernet traffic. In *Proceedings of the ACM SIGCOMM*.
- LIU, B., BRASS, P., DOUSSE, O., NAIN, P., AND TOWSLEY, D. 2005. Mobility improves coverage of sensor networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM, New York.
- MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M., AND SRIVASTAVA, M. B. 2001. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of ACM/IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA.
- PODURI, S. AND SUKHATME, G. S. 2004. Constrained coverage for mobile sensor networks. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*. IEEE Computer Society Press, Los Alamitos, CA.
- ROSS, S. M. 1996. *Stochastic Processes*. Wiley, New York.
- SHAH, R. C., ROY, S., JAIN, S., AND BRUNETTE, W. 2003. Data mules: modeling a three-tier architecture for sparse sensor networks. Tech. rep. IRS-TR-03-001, Intel Research.
- SUNDARESAN, A., VARSHNEY, P. K., AND RAO, N. S. V. 2007. Distributed detection of a nuclear radioactive source using fusion of correlated decisions. In *Proceedings of the Conference on Information Fusion (Fusion)*.
- WALDSPURGER, C. A., AND WEIHL, W. E. 1994. Lottery scheduling: Flexible proportional-share resource management. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.
- WANG, G., CAO, G., AND PORTA, T. L. 2004. Movement-assisted sensor deployment. In *Proceedings of ACM/IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA.
- WANG, W., SRINIVASAN, V., AND CHUA, K. C. 2005. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, New York.
- WANG, X., XING, G., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2003. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of ACM SenSys*. ACM, New York.
- ZHANG, H., AND HOU, J. C. 2005. Maintaining sensing coverage and connectivity in large sensor networks. *Wirel. Ad-hoc Sensor Netw.* 1, 1–2.
- ZHANG, X., KUROSE, J., LEVINE, B. N., TOWSLEY, D., AND ZHANG, H. 2007. Study of a bus-based disruption-tolerant network. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, New York.
- ZHAO, W., AMMAR, M., AND ZEGURA, E. 1994. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM, New York.
- ZOU, Y. AND CHAKRABARTY, K. 2003. Sensor deployment and target localization based on virtual forces. In *Proceedings of ACM/IEEE INFOCOM*. IEEE Computer Society Press, Los Alamitos, CA.

Received January 2009; revised September 2009 and December 2009; accepted February 2010