Matching and Fairness in Threat-based Mobile Sensor Coverage

by Chris Y. T. Ma, Jren-chit Chen, David K. Y. Yau, Nageswara S. Rao, Mallikarjun Shankar

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Matching and Fairness in Threat-based Mobile Sensor Coverage

Chris Y. T. Ma*, Jren-chit Chin*, David K. Y. Yau*, Nageswara S. Rao†, Mallikarjun Shankar†
*Department of Computer Science, Purdue University, West Lafayette, IN 47907.
†The Oak Ridge National Lab, Oak Ridge, TN.

***Abstract*—Mobile sensors can be used to effect complete coverage of a surveillance area for a given threat over time, thereby reducing the number of sensors necessary. The surveillance area may have a given *threat profile* as determined by the kind of threat, and accompanying meteorological, environmental, and human factors. In planning the movement of sensors, areas that are deemed higher threat should receive proportionately higher coverage. We propose a coverage algorithm for mobile sensors to achieve a coverage that will match – over the long term and as quantified by an RMSE metric – a given threat profile. Moreover, the algorithm has the following desirable properties: (1) stochastic, so that it is robust to contingencies and makes it hard for an adversary to anticipate the sensor's movement; (2) efficient; and (3) practical, by avoiding movement over inaccessible areas. Further to matching, we argue that a *fairness* measure of performance over the shorter time scale is also important. We show that the RMSE and fairness are in general antagonistic, and argue for the need of a combined measure of performance, which we call *power*. We show how a *pause time* parameter of the coverage algorithm can be used to control the tradeoff between the RMSE and fairness, and present an efficient offline algorithm to determine the optimal pause time maximizing the power. Lastly, we discuss the effects of multiple sensors, under both independent and coordinated operation. Analytical and extensive simulation results – under realistic coverage scenarios – are presented for performance evaluation.**

I. Introduction

A network of sensors can be used to protect people, livestock, or the environment against harmful substances in a geographical region. For example, Office of Naval Research (ONR) personnel have deployed a sensor network at the Port of Memphis to protect the area's population against the exposure to known chemical, biological, and radiological threats. A variety of sensor modalities is used to detect the presence of pollutants with sufficient accuracy and sensitivity. The project report [1] states that in choosing where to place the sensors, a pragmatic consideration is to select locations that are accessible. Besides accessibility, the report concludes that the primary factor in deciding the placement of a limited quantity of sensing resources is to assess its impact on the area's population distribution, since "human effects represent the true consequences of failure" to detect a harmful agent and subsequently evacuate the affected population. It states that densely populated areas should receive priority attention relative to unpopulated or sparsely populated areas. Moreover, historical meteorological information, such as *wind rose* data characterizing the predominant seasonal distribution of wind speeds/directions, should be considered. This is because the spread of a chemical/biological/radiational plume is affected by wind conditions, and sensors in the wind's direction will be able to monitor the most vulnerable areas and detect the plume with the smallest delay.

The Memphis Port deployment exemplifies the need and benefits to provide differential sensor coverage of different geographical areas based on a concept of *threat level*. Intuitively, the threat level of an area quantifies the relative danger of exposing the area due to non- or under-coverage. An area may have a high threat level because it is under high risk or because a realized risk in the area will produce severe consequences. In the Memphis Port experiment, static sensors are used. Because of the limited number of sensors, the Port area cannot be fully covered [1]. In this paper, we consider the use of mobile sensors to cover a whole surveillance area over time. Because the area coverage occurs over time, and does not have to be complete all the time, a significantly smaller number of sensors can be used compared with static sensors. We assume that the movement of a sensor can be under program control. For example, the sensor is carried by a robot supporting programmable movement. We then consider the design of a mobility algorithm to control the sensor's movement, such that it can effect a coverage profile that matches a given threat profile.

It is clear that the economical savings of using fewer sensors have to be balanced against the costs of supporting the movement. We do not attempt to answer the question of economic tradeoff definitively one way or the other, as it depends in part on the difficulty of the sensing task relative to the movement task and in part on future technological developments. We do notice, however, that commodity robots [2], for instance, are available that are rather inexpensive, and believe that it is interesting to explore such a tradeoff. Moreover, there may be other reasons to use mobility besides economics. For example, robots may be used because they can carry sensors over

Research was supported in part by the Office of Naval Research under grant number DE-AC05-00OR22725.

a deployment field that is hard to get to for installing a static sensor (e.g., an undersea environment or the accessibility placement condition of the Memphis Port deployment). Also, mobility can be more robust against an unplanned sensor failure (e.g., an area left uncovered by a failed sensor can be covered later by another sensor that moves into the area) or an unexpected contingency (e.g., an obstacle unexpectedly appears that obstructs the sensing path, and a mobile sensor is able to navigate around the obstacle).

Our contributions in this paper are as follows:

- To the best of our knowledge, this paper represents a first effort to investigate general threat-based coverage by sensors that move during deployment.
- We identify *matching* and *fairness* as the major performance criteria in evaluating the effectiveness of coverage. We show that the two performance measures are in general antagonistic, and discuss their tradeoffs. We show how the two metrics can be unified to give a combined metric of *power*, by adopting a common view of utility.
- We present the development of a coverage algorithm for sensors to achieve effective matching and fairness simultaneously under realistic deployment scenarios. The algorithm provides a tunable parameter to control the tradeoff between matching and fairness. The optimal parameter maximizing a given power metric can be computed by an efficient offline algorithm.
- We present analytical and simulation results to quantify the performance of our algorithms. In particular, our main results show that
  - A complement of techniques contribute to a coverage algorithm that can match the coverage profile of a mobile sensor to a given threat profile with excellent accuracy.
  - There is an inherent tradeoff between matching and fairness.
  - Using more sensors can significantly improve the fairness of coverage, although the marginal improvement due to an additional sensor decreases slightly as the number of sensors increases.
  - Using more sensors – either independently or under basic coordination – does not significantly improve the accuracy of matching. Rather, the matching deteriorates slightly due to possible redundancy of coverage by multiple sensors.
  - For multiple sensors, basic coordination approaches do not improve performance over independent operation. Rather, independent operation of the sensors in stochastic movement is viable, because it is simple, is shown to be effective, and is robust to sensor failure.

II. Related Work

Static sensor coverage. A significant amount of research has studied the placement problem of static sensors for optimal area coverage [3], quality of surveillance [4], or energy efficiency [5]. The relationship between coverage and connectivity, given communication range c and sensing range s, has been studied for the cases of $c/s \leq 1$ [6] and $c/s < \sqrt{3}$ [7]. The generalization to any c/s and to 2 k-connectivity is given in [8]. The tracking of a moving target by a network of static sensors has been studied in [9], [10]. In [9], a protocol is presented to enable the sensors to transition to a low-power state to conserve energy, without compromising the quality of surveillance. In [10], the number of sensors needed to track a raditional source under uncertainty is analyzed. The strategy for a moving *target* to evade a static network of sensors with minimum exposure is discussed in [11].

In the area of static sensor placement, the work closest to our problem is the Memphis Port sensor network deployment [1]. They present an iterative algorithm to place a given number of n sensors around the Port of Memphis to maximize the protection of the area's population. At each step of the algorithm, they use a search procedure to place the next sensor at a position that will maximize the marginal gain in risk coverage. Our work addresses similar threat-based coverage, but in the context of mobile sensors.

Mobile sensor coverage. Previous work on sensor mobility has focused on moving the sensors to deployment locations that optimize the area of coverage [12], [13]. The sensors do not move during the sensing task. Other hybrid mobile/static networks have used moving relay nodes to collect data from static information sources [7]. They show that using the mobile relay as the sink is the most efficient, and that, for a dense network, the improvement in network lifetime of one relay is upper bounded by a factor of four over a static network. In [14], optimal algorithms are presented to move the sink adaptively according to the flow of current events, to minimze the communication energy or the maximum load on a specific sensor.

Due to trends in robotics and embedded sensor technologies in vehicles, there is a growing amount of work on sensors that move during deployment. The area coverage of mobile sensors is characterized in [15] under the assumption of a uniform node density. They show that mobility can significantly reduce the number of sensors needed to detect a randomly located stationary target in a given amount of time. If the target can also move and is intelligent, it can plan its movement to avoid detection. In that case, a pursuit-evasion game can be defined. A greedy policy for directing a group of moving agents to "swarm" locations with the highest probabilties of finding an evader is proposed [16], and is shown to find an evader in finite time. The implementation of the theoretical game on unmanned aerial/ground vehicles is discussed in [17], and the use of mobile sensors orbiting in space to help minimize the time-of-capture of the evaders is considered in [18].

In the area of mobile sensors during deployment, the work closest to our problem is [19]. In their work, they study the capture of *transient* events by a mobile sensor. The events arrive/depart according to given stochastic processes at given points of interest (PoI) in a circular space. They show how a sensor moving in a circle at variable speeds can optimize its movement to detect the largest fraction of events. Their PoIs can be viewed as a specialized threat profile, since they are the positions where interesting targets are likely to appear compared with non-PoIs. They use a variable speed but restrict the path to be circular around what is essentially a one-dimensional space, whereas we use a fixed speed but randomness in the path selection over a 2D space.

III. Problem Formulation

We consider the surveillance of a network area, which we call the *map*, for a given threat by one or more mobile sensors. For simplicity, we assume that the map is a two-dimensional rectangular space, of dimensions $x \times y$, where x and y are in distance units. The map is partitioned into an $m \times n$ array of cells, each of dimensions $S \times S$ (in distance units), and S divides both m and n. The cells are enumerated by their unique integer ids, $0, 1, \ldots$, in top-to-bottom row order, and left-to-right column order within each row.

The distribution of threat in the map area is characterized by a *threat profile*, denoted by Φ. The threat level of a cell, say i, is given by $\Phi(i)$, and quantifies the risk of not covering the cell relative to the other cells. As motivated in Section I, the threat profile should be determined according to the application, namely the kind of threat, and any relevant meteorological, environmental, and human factor. In addition, we allow certain cells to be marked as *inaccessible*, meaning that a sensor can neither monitor nor travel over such a cell due to physical limitations or policy decisions. An inaccessible cell, say i, has a threat level of $\Phi(i) = \text{NaN}$. Mathematically, Φ is a probability distribution: $0 \leq \Phi(i) \leq 1, \{\forall i : \Phi(i) \neq \text{NaN}\}$ and $\Sigma_{j:\Phi(j)\neq \text{NaN}} \Phi(j) = 1$.

In solving the coverage problem, areas that are deemed higher threat should receive priority attention in the form of proportionately higher coverage. The goal is achieved by a mobility algorithm that controls the movement of the sensors (see Section IV). As a sensor moves, it will enter different cells. For the purpose of bookkeeping, we assume that a cell is *covered*, in the sense that any threat event present in the cell is detected, whenever a sensor is inside the cell. By bookkeeping on a per-cell basis, the bookkeeping costs can be kept low, although there may be some loss of precision in the matching. However, the precision loss is small if the cell dimension S is comparable to the sensing range of the sensor. Moreover, the assumption is not needed in actual operation, and is not used in the algorithm design in Section IV. The fraction of time that a sensor, say l, spends in each cell up to time t is given by the sensor's *coverage profile*, denoted by Π_t^l. Specifically, $\Pi_t^l(i)$ gives the fraction of time that the sensor l spends in the cell i up to time t. Similar to Φ, Π_t is a probability distribution. When the context of the sensor is clear, we drop the superscript l for simplicity.

For simplicity, we assume that the threat profile is time invariant. In practice, it is clear that when the threat profile changes, we can use the current profile as a new input to the mobility algorithm, and the sensor's mobility will adapt accordingly. Threat profiles in real life are likely to depend on factors that are close to static because they change slowly – e.g., population, locations of strategic facilities, and seasonal changes of weather.

Performance measures. For a mobile sensor, the **matching** between the given threat profile and the achieved coverage profile at time t is quantified by the following root mean square error (RMSE) measure:

$$RMSE_t = \sqrt{\frac{\sum_i \left(\Phi(i) - \Pi_t(i)\right)^2}{m \times n}}.$$

If the sensor's movement is a stationary stochastic process, the coverage profile will reach a steady state distribution, and the limit $lim_{t\to\infty}\Pi(t)$ exists, which will in turn determine the steady state matching performance of the algorithm.

The matching measure alone does not fully evaluate the performance of a coverage algorithm. Consider the monitoring of a cell whose threat level is 0.1. A coverage algorithm may achieve a 10% coverage of the cell in the steady state, but does so by spending one month in the cell once every 10 months. The average *exposure time* of the cell, i.e., the average duration of the continuous time interval over which the cell is not covered, is 9 months, and would be unacceptable if, say, the application is to monitor a residential area for flooding. In contrast, another algorithm that visits the cell (i.e., the residential area) for one minute every 10 minutes achieves the same 10% coverage, but never leaves the cell uncovered for more time 9 minutes. Any flood event can then be detected and reported in a timely manner. To further quantify the time scale over which a certain matching is achieved, we define an **unfairness** measure, denoted by $\overline{\mathcal{F}}$, of the algorithm, as follows:

$$\overline{\mathcal{F}} = \sum_i \epsilon(i) \times \Phi(i),$$

where $\epsilon(i)$ is the average exposure time of cell i. Notice that the unfairness is defined as the weighted average of the exposure time of each cell by the threat of that cell, and is a time quantity. We therefore assess fairness by its dual unfairness measure. A fair algorithm is then one that achieves a low unfairness. For a persistent threat event, i.e., an event that remains present once it appears, the unfairness measures the weighted average of the delay until the event is detected after occurrence. For the transient events discussed in [19], both the unfairness and the matching determine the weighted average fraction of the events that will be missed.

IV. Mobility Algorithms

In this section, we develop algorithms for a mobile sensor to determine its movement and effect coverage that matches a given threat profile with high accuracy. We target the following desirable properties of the algorithm in our design:

- *Accurate.* The algorithm should achieve a low RMSE of coverage against the threat profile (see Section III).
- *Fair.* The algorithm should be fair (i.e., have a low unfairness value) in the sense of Section III.
- *Stochastic.* The random movement makes it hard for an adversary to anticpate the sensor's movement and hence avoid detection. Also, random movement enables n sensors to be deployed independently without advance schedule planning or runtime coordination, but still with good performance benefits (as shown in Section VIII).
- *Efficient.* The algorithm should have low space and time complexities, so that it can be efficiently executed on the mobile sensor.
- *Practical.* The algorithm should admit and obey given *accessibility constraints* for the coverage area. For example, a sensor carried on a terrestrial vehicle will not be able to enter sea areas in a geographical region. The algorithm should avoid movement over the inaccessible areas.

As a starting point of our design, we use a *weighted random waypoint* (WRW) algorithm. The random waypoint formulation in [20] has been used widely to *model* user/device movement in a mobile network, in which case there is a significant debate about whether the model is realistic or not. Notice that the concern of realism does not apply in our problem context, since our objective is not to model a mobile network, but to develop an *algorithm* for determining the sensor movement. In our algorithm, a sensor moves in a sequence of *trips*. The tth trip, $t = 0, 1, \ldots$, starts at a (uniformly) random position in cell s_t and ends at a (uniformly) random position in cell d_t, the $(t+1)$st trip starts at a random position in cell $s_{t+1} = d_t$ and ends at a random position in cell d_{t+1}, and so on. We pick a random position with uniform probability inside a cell, because each cell is of non-negligible dimensions $S \times S$. For simplicity, we assume henceforth that when we say a trip starts/ends at a cell, it is understood that the trip starts/ends at a random position inside the cell. The movement from s_t to d_t occurs in a direct, straightline path at speed v_t. In standard random waypoint [20], the speed v_t is selected uniformly randomly from a range $[v_{min}, v_{max}]$, and each destination d_t, also called a *waypoint*, is selected uniformly randomly from the whole map. In our algorithm, we let the sensor move at a fixed speed v specified for that sensor. Moreover, to be threat-aware, our algorithm will consider the given threat profile in choosing a waypoint, and select a cell i as the waypoint with probability Φ_i, which is the threat level of i.

The WRW algorithm is simple and probabilistic, thus meeting the third and fourth design objectives. Moreover, it attempts to achieve a coverage that matches the threat profile, by considering the profile in selecting the waypoints. The basic algorithm, however, fails to achieve an accurate match because it fails to consider the effects of a trip on covering the *intermediate* cells between the source and destination. For example, consider a map with a few high threat *hotspots*. In attempting to move between the hotspots to give them sufficient coverage, the sensor will also visit frequently all the cells between the hotspots, thereby overcovering the intermediate cells. The analytical result in Section VII gives a more complete characterization of the WRW algorithm. To overcome the weaknesses of the basic algorithm, WRW can be used in conjunction with the following complementary techniques:

Maximum trip length. In this variation, we do not allow the distance of a trip to exceed a given parameter L (in distance units). Hence, in choosing the next waypoint d_{t+1} after the tth trip, we constrain the candidate cells to be within the disc centered at d_t of radius L. The choice of the waypoint among the restricted set of candidate cells occurs as in the basic algorithm. Limiting the trip length helps to decouple the intermediate cells visited from a set of high threat cells that require frequent visits. For example, consider two hotspots, say i and j, in a map. A suitable maximum trip length will force the sensor to consider more possible paths to move between i and j, thereby reducing the possibility of "warming up" the intermediate cells as a side effect.

Adaptivity to prior coverage. Because of the stochastic nature of the WRW algorithm, and the correlations between cells visited due to their physical positions, the algorithm's actual coverage at any point in time may deviate from the given threat profile. To avoid such deviations from accumulating to an unacceptable level, we propose to use the sensor's prior coverage as an input in selecting the next waypoint. Specifically, we compute the *undercoverage* of each cell, say i, as

$$\overline{C}_t(i) = \max\{0, \Phi(i) - \Pi_t(i)\}$$

where $\Pi_t(i)$ is the fraction of time that cell i was visited by the sensor up until the end of the tth trip. Then, the probability that a candidate cell, say i, is chosen as the next waypoint d_{t+1} is proportional to $\overline{C}_t(i)$. Considering undercoverage as a selection criterion has the obvious advantage of ramping up visits to cells that have been neglected relative to their threat level, at the expense of cells that have received too much prior coverage.

Random pause time. To raise the coverage of an undercovered cell, say i, in order to improve matching with the threat profile, the most efficient approach is for the sensor to stay in i for long enough to correct the undercoverage. The approach is extremely efficient because it requires zero overhead of movement and there is no possibility of inadvertently changing the coverage of other cells due to the (now avoided) movement. However, by staying at the current cell longer, clearly the sensor will take longer before it can return to a previously visited cell. Hence, fairness suffers, showing that there is an inherent tradeoff between improving matching efficiently/accurately and being fair. The issue is not unlike scheduling in traditional systems areas. For example, in CPU scheduling, improving fairness requires increased context switching between processes, which reduces the efficiency of the global system. To enable a useful and controllable tradeoff between the RMSE and unfairness metrics, the sensor, on reaching the destination of a trip, will stay at the destination for a *pause time* p (in time units) before selecting the next waypoint. The time p is drawn randomly from a distribution determined by a pause time parameter denoted by P (in time units). Specifically, at the end of the tth trip at destination cell i, $p \sim \mathbf{U}(0, \Omega_t(i))$, where

$$\Omega_t(i) = \frac{P \times \Phi_t(i)}{\Sigma_{j \in \mathcal{C}} \Phi_t(j)}$$

for the basic WRW algorithm, and

$$\Omega_t(i) = \frac{P \times \overline{C}_t(i)}{\Sigma_{j \in \mathcal{C}} \overline{C}_t(j)}$$

for the WRW variant that is adaptive to prior coverage, and $\mathcal{C}$ is the set of cells that are candidates as the next waypoint.

Family of algorithms. Notice that the complement of features augmenting the WRW algorithm can be orthogonally combined, thereby offering a *family* of algorithms for threat-based mobile coverage. We will denote a particular augmented algorithm by WRW-$feat$, where $feat$ is a list of letters enumerating the augmentations in alphabetical order, and the letters L, a, and P, are for the “maximum trip length”, “adaptivity to prior coverage”, and “random pause time” features, respectively. For example, WRW-L denotes the WRW algorithm with the maximum trip length constraint, and WRW-aLP denotes the algorithm with all the three features enabled. The experimental results in Section VIII show that each feature contributes positively to accurate matching, and hence the WRW-aLP algorithm is the most powerful in the matching respect. Additionally, the pause time parameter in WRW-aLP enables a useful tradeoff between matching and fairness, an issue that we will address in Section IV-A.

```
SelectWaypoint $(L, P, \overline{C}_t, x_t)$
  Initialize $u \sim \mathbf{U}\,(0,1)$, $a := 0$, $b := 0$, $c := 0$;
  For each cell $i$ within range $L$ of $x_t$
        If (Accessible($x_t, i$))
          $a := a + \overline{C}_t(i)$;
          If $b < \overline{C}_t(i)$
            $b := \overline{C}_t(i)$;
        Endif
  Endfor
  For each cell $i$ within range $L$ of $x_t$
        $c := c + \overline{C}_t(i)/a$;
        If $u < c$
          pick $x_{t+1}$ as random point inside $i$
            with uniform probability;
          $p = \overline{C}_t(i)/(a \times b)$;
        Endif
  Endfor
  return $(x_{t+1}, p)$;

WRW-aLP($\Phi, L, P$)
   While(true)
        $(x_{t+1}, p) :=$ SelectWaypoint$(L, P, \overline{C}_t, x_t)$;
        move to $x_{t+1}$;
        pause for $p$ time;
        update $\overline{C}_t(i)$;
   Endwhile
```

Fig. 1. Specification of WRW-aLP algorithm.

The WRW-aLP algorithm is specified in Fig. 1. In the specification, the **WRW-aLP** program takes as input the threat profile Φ, and the L and P parameters of the WRW-aLP algorithm. The function **SelectWaypoint**, takes four input parameters, in which the parameter x_t is the current position of the sensor, and returns the destination and pause time of the next trip. The **Accessible** function (whose specification is not shown) checks if all the intermediate cells connecting a given pair of cells are accessible, and can be precomputed for each given pair. Either for-loop in **SelectWaypoint** has a complexity of $O(L_s^2)$, where $L_s = L/s$. Hence, WRW-aLP requires $O(L_s^2)$ computation after every trip of length $O(L)$. The space costs of storing either the map of cells or the precomputed **Accessible** function is $O(m \times n)$. Hence, WRW-aLP can handle given inaccessibility constraints and has an efficient implementation. The experimental results in Section VIII evaluate the algorithm’s effectiveness in also matching the fairness and accuracy objectives.

A. Matching, Fairness, and Power

The dual concerns of matching and fairness means that coverage algorithms must be compared in a two-dimensional space. Moreover, the inherent tradeoff between the two concerns means that it will be impossible to rank many interesting algorithms in a total order. Rather, in comparing two algorithms, say A and B, A may perform better in one respect, but less well in the other. Whether A or B is preferred in a given situation should depend on the context of the situation, such as the preferences of the user, or the characteristics of the application. We seek an approach to rank algorithms by a single, unifying metric, after appropriately considering the specifics of the situation.

The major difficulty in unifying the two metrics is that they are of completely different natures: Matching is measured as an RMSE, which is a percentage quantity, whereas unfairness is the threat-weighted average exposure time, a time quantity, between successive visits to the same cell. How do we combine a percentage value and a time quantity, while addressing the issue of user preferences? Our approach recognizes that a user, in the context of a given application, derives a certain level of "satisfaction" from an achieved level of performance, in either performance aspect. For example, in monitoring a residence for flooding, the user may be quite satisfied (i.e., have a 100% level of satisfaction) if each room is checked at least every two hours, on average, but is completely dissatified (i.e., have a zero level of satisfaction) if a room may be left unchecked for a whole day, on average. Between two hours and 24 hours, the user's level of satisfaction decreases linearly from one to zero. The example can be expressed as a *utility function,* $\mathcal{U}_{\mathcal{F}}(\cdot)$, similar to the one in Fig. 6, where the utility, a number between zero and one, is given as a function of the unfairness of coverage. A similar utility function for matching, denoted by $\mathcal{U}_{\mathcal{M}}(\cdot)$, characterizes the utility as a function of the achieved RMSE.

After mapping both RMSE and unfairness values to utility quantities, we can define the *power* of an algorithm as a weighted sum of the utilities; i.e.,

$$power(f, m) = \alpha \times \mathcal{U}_{\mathcal{F}}(f) + (1 - \alpha) \times \mathcal{U}_{\mathcal{M}}(m)$$

where f and m are, respectively, the unfairness and RMSE achieved by the algorithm, and $0 \leq \alpha \leq 1$ expresses the importance of fairness relative to matching in the deployment context. Henceforth, we will say that an algorithm has a better performance if it has a higher power than another algorithm.

V. Finding Optimal Pause Time Parameter

Characterizing the power as a function of the pause time parameter P is extremely difficult and expensive, as it would require solving the steady-state RMSE and unfairness values over the whole range of interest of P. To efficiently compute the optimal P that maximizes the power, without completely knowing the power function, we use Brent's method [21].

Brent's method is a one dimensional optimization method that does not require the derivative of the objective function. It is suitable for our problem because (1) characterizing the power function is expensive, and (2) the derivative of the power function may not exist.

To apply Brent's method, we first need to find two abscissa a and b that bracket the optimal P. To do so, we start with a zero P and magnify the bracket by the golden ratio until we overshoot the optimal input of the power function. During the magnification, we also shift the bracket to eliminate intervals that are known not to contain the optimal point. Given the two abscissa a and b, we then compute the optimal point c within a and b. Two methods of searching for the optimal point are used: (1) the inverse parabolic interpolation technique, and (2) the golden section search. The first method is more efficient and converges faster than the second method or a linear interpolation technique. However, it may not always succeed for certain types of function. Therefore, we fall back on the golden section search in the case that the inverse parabolic interpolation fails to produce a solution.

VI. Multiple Sensors

When the surveillance area is large, one sensor may not be sufficient to cover the area with good performance. Increasing the speed of the sensor may help to some extent, in that the sensor can move more quickly between cells that require attention and apportion its service more efficiently. A more effective approach is, however, to fundamentally increase the amount of sensing resources available by deploying n sensors.

The most simple strategy of deploying n sensors is to deploy them independently, each working according to the WRW-aLP algorithm. The stochastic property of the algorithm makes it likely for these sensors to distribute their service well over the network area, even without advance schedule planning or close coordination at run time. There is a concern, however, if the number of sensors is large relative to the size of the surveillance area. As the area becomes relatively smaller, it is more likely for the sensing ranges of the sensors to overlap. The overlapped coverage is wasteful because one sensor would be sufficient to detect a threat event by our problem formulation. As a result, we also investigate two preliminary approaches of deploying multiple sensors in a coordinated manner:

- **Knowledge of global coverage profile.** In this approach, we assume that each sensor knows, at time t, the fraction of time that a cell is covered by *any* sensor up to time t. In adapting to the effects of the prior coverage, each sensor will then determine, independently but based on the undercoverage of each cell by *all* the sensors, the cells that should receive priority attention in the future coverage. Two observations are in order. First, although the sensors use the global coverage history as information, they will not communicate in order to avoid visiting the same cell at the same time. Hence, redundant coverage is not eliminated. Second, there is clearly a need to disseminate the coverage information of individual sensors to the global network in an implementation of this approach. The information exchange can be readily supported if we assume, for example, the existence of a cellular phone infrastructure and the sensors are equipped with the necessary cellular communication interfaces. Nevertheless, our goal in this paper is not to consider how such information exchange should occur nor its runtime overhead. Instead, we are interested in the benefits of having the global information assuming that it is available.

- **Static division of responsibilites.** This approach seeks to eliminate the redundancy of coverage by partitioning the responsibilities for covering different cells between the sensors in a disjoint but complete manner. In essence, each sensor, say i, is assigned a *job* as a connected set of cells, denoted by J_i, such that $J_i \bigcap J_j = \phi$ for $i \neq j$, and $\bigcup_i J_i = \{\text{full set of accessible cells}\}$. Each sensor then uses the WRW-aLP algorithm to cover its set of cells in a threat-based manner. There are different ways to perform the partitioning into jobs. In particular, the *division by equal area* produces jobs that have an equal area of cells that are accessible. The *division by equal threat* produces jobs whose cells have the same aggregate threat level. In general, multiple actual divisions exist that can achieve either objective.

VII. Analytical Result

We present the analytical property of the basic WRW algorithm.

Theorem 7.1: Let $\mathcal{M}$ denote a surveillance map. The steady state coverage distribution of the WRW algorithm is given by

$$\Pi\left(i\right) = \frac{\sum\limits_{p\in\mathcal{M}}\sum\limits_{q\in\mathcal{M}} \Phi\left(p\right)\Phi\left(q\right)\mathcal{T}_{p\to q}\left(i\right)}{\sum\limits_{i'\in\mathcal{M}}\sum\limits_{p\in\mathcal{M}}\sum\limits_{q\in\mathcal{M}} \Phi\left(p\right)\Phi\left(q\right)\mathcal{T}_{p\to q}\left(i'\right)}$$

where

$$\mathcal{T}_{p\to q}\left(i\right) = \frac{\|(p\to q)\cap i\|}{v}$$

is the expected sojourn time in cell i during a single trip from p to q.

Proof: Let $s_t \in \mathcal{M}$ and $d_t \in \mathcal{M}$ denote the starting and ending cells of the trip t. In the WRW algorithm, the mobile sensor selects the next waypoint d_{t+1} according to the threat profile Φ upon arriving at d_t. It is clear that the waypoint selection process is an order-0 Markov process, with the waypoint distribution after the tth trip given by $\Psi_t\left(i\right) = \Phi\left(i\right)$.

Let $\mathcal{T}_n\left(i\right)$ denote the total sojourn time in cell i after n trips. $\mathcal{T}_n\left(i\right)$ can be written as the sum of sojourn times in cell i for all the trips $0,\ldots,n-1$. We have

$$\mathcal{T}_n\left(i\right) = \mathcal{T}_{s_0\to d_0}\left(i\right) + \mathcal{T}_{s_1\to d_1}\left(i\right) + \ldots + \mathcal{T}_{s_{n-1}\to d_{n-1}}\left(i\right)$$

Hence,

$$\begin{aligned}\mathrm{E}\left[\mathcal{T}_n\left(i\right)\right] &= \mathrm{E}\left[\mathcal{T}_{s_0\to d_0}\left(i\right) + \ldots + \mathcal{T}_{s_{n-1}\to d_{n-1}}\left(i\right)\right]\\ &= \mathrm{E}\left[\mathcal{T}_{s_0\to d_0}\left(i\right)\right] + \ldots + \mathrm{E}\left[\mathcal{T}_{s_{n-1}\to d_{n-1}}\left(i\right)\right]\\ &= \sum_{p\in\mathcal{M}}\sum_{q\in\mathcal{M}} \Psi_0\left(p\right)\Psi_1\left(q\right)\mathcal{T}_{p\to q}\left(i\right) + \ldots\\ &+ \sum_{p\in\mathcal{M}}\sum_{q\in\mathcal{M}} \Psi_{n-1}\left(p\right)\Psi_n\left(q\right)\mathcal{T}_{p\to q}\left(i\right)\end{aligned}$$

By the Markov property, we have

$$\mathrm{E}\left[\mathcal{T}_n\left(i\right)\right] = \sum_{i=0}^{n-1}\sum_{p\in\mathcal{M}}\sum_{q\in\mathcal{M}} \Phi\left(p\right)\Phi\left(q\right)\mathcal{T}_{p\to q}\left(i\right)$$

The steady-state coverage profile of the WRW algorithm Π is then

$$\begin{aligned}\Pi\left(i\right) &= \frac{\lim_{n\to\infty}\mathrm{E}[\mathcal{T}_n(i)]}{\lim_{n\to\infty}\sum\limits_{i'\in\mathcal{M}}\mathrm{E}[\mathcal{T}_n(i')]}\\ &= \frac{\sum\limits_{p\in\mathcal{M}}\sum\limits_{q\in\mathcal{M}}\Phi(p)\Phi(q)\mathcal{T}_{p\to q}(i)}{\sum\limits_{i'\in\mathcal{M}}\sum\limits_{p\in\mathcal{M}}\sum\limits_{q\in\mathcal{M}}\Phi(p)\Phi(q)\mathcal{T}_{p\to q}(i')}\end{aligned} \qquad (1)$$

■

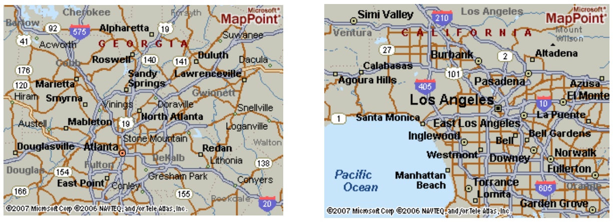
VIII. Simulation Results

We report simulation results to illustrate the performance of our algorithms. We consider coverage of a number of metropolitan cities, including San Francisco, Los Angeles (LA), Atlanta, Paris, London, and Tokyo. The boundary longitudes/latitudes of the cities and the sizes of their populations are shown in Table I. The maps of Atlanta and LA are also shown in Fig. 2. As formulated in Section III, each city area is divided into a two-dimensional grid of cells. The division is according to the LandScanTM 2004 database of global population data [22]. LandScanTM provides population data in a cellular grid format with each cell corresponding to 1/120 degree of longitude in width and 1/120 degree of latitude in height. For ease of interpretation, we project the LandScan data to cartesian coordinates, according to World Geodetic System ’84 (WGS84). The projection gives us $s \approx 0.75$ km, or a cell size of about 0.75 km × 0.75 km. Due to space constraints, we present selected experimental results in this section. The presented results are representative.

We assume that terrestrial mobile sensors are used over the cities to monitor, say, air pollutants with health impact on people. Hence, the threat level of a cell is defined as the size of the population inside that cell, because a more densely populated area will endanger more people if left uncovered. Since we model terrestrial sensors, water areas in a map (e.g., the Pacific Ocean part of the LA map) are defined to be inaccessible. In the maps used in our experiments, the water areas do

City	Northeast		Southwest		Average	Height	Dimension	Population
	Latitude	Longitude	Latitude	Longitude	Width (km)	(km)		
Atlanta	34.033333	-84.025000	33.700000	-84.616667	54.75	36.97	40 × 70	1982086
London	51.600000	0.100000	51.400000	-0.308333	28.35	22.25	24 × 48	4228314
Los Angeles	34.195833	-118.120833	33.895833	-118.570833	41.55	33.28	36 × 54	4599286
Paris	49.029167	2.687500	48.729167	2.012500	49.51	33.36	37 × 81	8449465
San Francisco	37.820833	-122.304167	37.687500	-122.545833	21.30	14.80	16 × 29	802056
Tokyo	35.812500	139.962500	35.545833	139.504167	41.49	29.59	32 × 55	12072968

TABLE I
POSITION AND POPULATION DATA OF SIX CITIES.



(a) Atlanta, GA

(b) Los Angeles, CA

Fig. 2. Maps of the cities under surveillance.

not partition the land areas. Hence, it is possible for one sensor to cover all the land areas given enough time.

Parameters and performance measures. We use the performance measures of matching and unfairness as defined in Section III. For matching, we scale the RMSE by the population size of the city, which gives a mismatch measured in number of people. For the unfairness, we report the weighted average exposure time in *time units*, where each time unit is 180 seconds. Unless otherwise specified, the following parameters are used in the experiments: (1) A mobile sensor moves at a speed of $3S$ / time unit, or about 34.8 mph; and (2) Where applicable, the maximum trip length parameter is set to be $L = 10 \times S$. Results are reported as averages of 50 simulation runs. The 25- and 75-percentiles are reported in certain experiments, in which case they are shown to be close to the means, and are omitted in the other experiments because of their small deviations from the means.

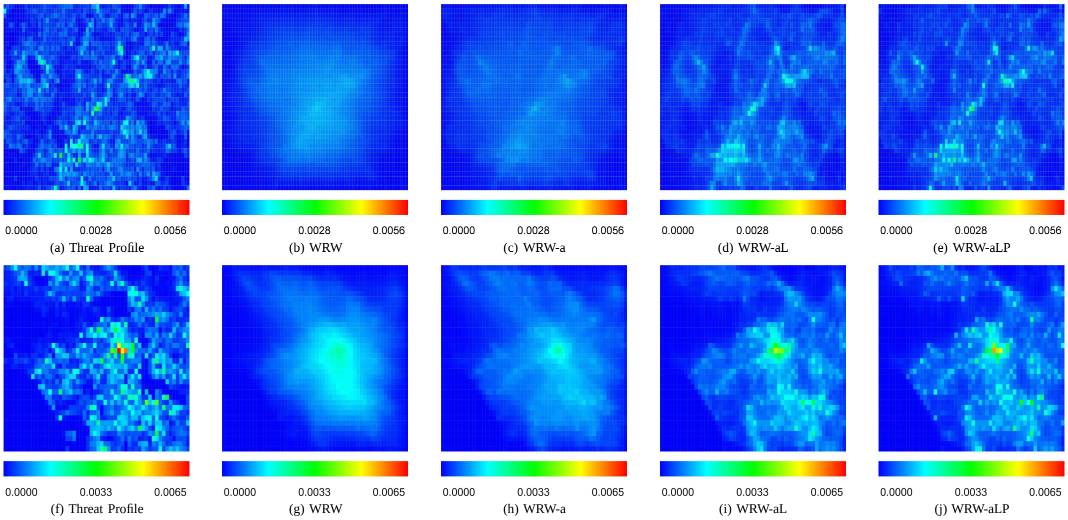
A. Matching by one sensor

In this experiment, we use one sensor to cover a city area, using various instances of the WRW family of algorithms in Section IV. Because the cities are large, it takes one sensor a significant amount of time to cover an entire city area. In particular, the unfairness numbers are of the order of several hours, which represent an inherent limitation due to constrained physical resources, and not due to the coverage algorithms. As the results in Section VIII-D show, the unfairness can be significantly decreased by using multiple sensors. Nevertheless, the results in this section illustrate the major performance properties of the coverage algorithms.

Fig. 3(a) and Fig. 3(f) give the threat profiles of Atlanta and LA, respectively. Figs 3(b)–(e) show the achieved steady-state coverage profiles of the WRW, WRW-a, WRW-aL, and WRW-aLP algorithms, respectively, for Atlanta. Figs 3(g)–(j) show the corresponding achieved steady-state coverage profiles for LA. For the WRW-aLP algorithm in this experiment, the pause time parameter is set to be one time unit. Visually, the matching with the threat profile improves as we progress from Fig. 3(b) to Fig. 3(e), or from Fig. 3(g) to Fig. 3(j). The visual observation can be quantitatively confirmed by referring to Fig. 4(a), in which we show the RMSE achieved by each algorithm normalized to the RMSE of the WRW algorithm (i.e., the RMSE of the WRW algorithm is shown as one, and the normalized RMSE of each algorithm shows the algorithm's percentage improvement over WRW.) For the five cities shown, including Atlanta and LA, the normalized RMSE consistently decreases from left to right. Additionally, Fig. 4(b) shows the unfairness of each algorithm normalized to the unfairness of the WRW algorithm. Observe that the unfairness numbers of WRW-a, -aL, and -aLP are about the same, and are significantly smaller than the WRW unfairness. *We conclude that the progression of features, namely, a, aL, and aLP, each contributes to increased matching accuracy, and WRW-aLP is the most powerful algorithm in the matching respect. Moreover, the more accurate matching is achieved without hurting the fairness.*

B. Impact of pause time parameter

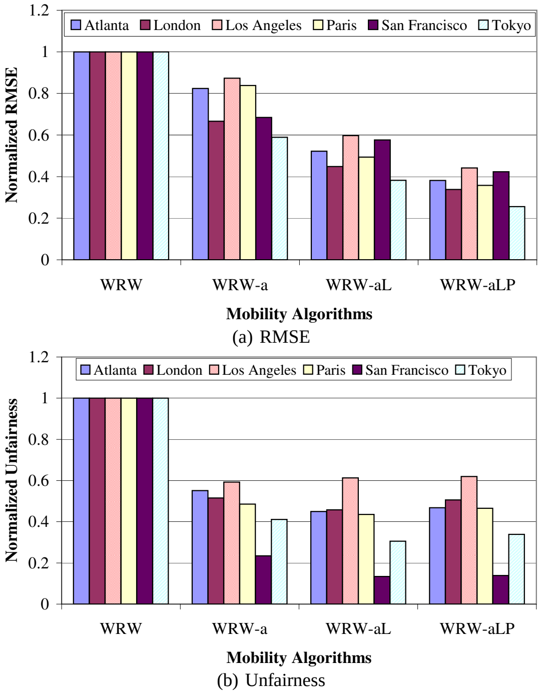
We illustrate the impact of the pause time parameter P on the matching and unfairness measures, for the case of one mobile sensor using the WRW-aLP algorithm. In this set of experiments, we vary the pause time paramter to be $P = 1$, 2, 4, 8, 16, 32, and 64 time units. Fig. 5 show combined plots of the RMSE (left y-axis) and the unfairness (right y-axis) as a function of P, for Atlanta and LA. Notice that for both figures, as the pause time increases, (1) the unfairness increases, in a partly constant, partly linear manner; and (2) the RMSE decreases like $1/(P + c)$, where c is a small constant. We also show the 25- and 75-percentiles of the RMSE in Table II for the set of runs for Atlanta. Notice that the values deviate little from the averages. We will omit the



(a) Threat Profile (b) WRW (c) WRW-a (d) WRW-aL (e) WRW-aLP

(f) Threat Profile (g) WRW (h) WRW-a (i) WRW-aL (j) WRW-aLP

Fig. 3. Threat profiles and steady-state coverage profiles of mobility algorithms for Atlanta (a)–(e) and LA (f)–(j).



(a) RMSE

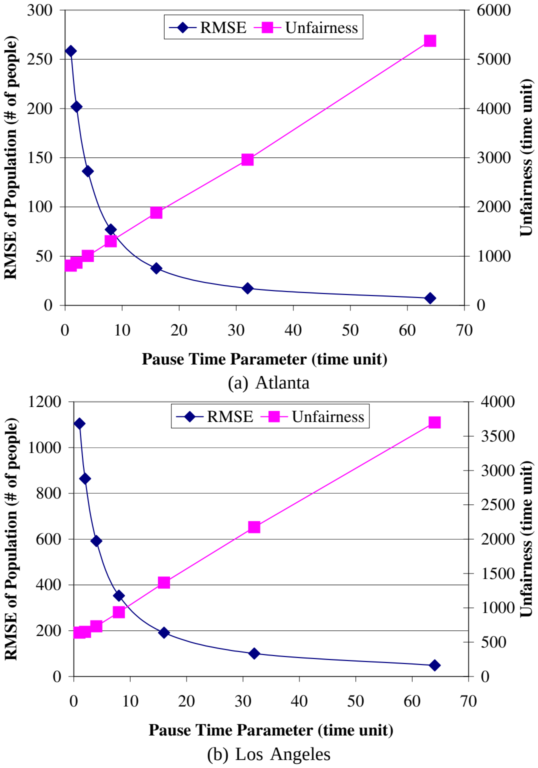
(b) Unfairness

Fig. 4. RMSE and unfairness of mobility algorithms for six cities.

25- and 75-percentiles of the data distributions for the future sets of experiments, due to their closeness to the means. From this set of experiments, *we conclude that there is an inherent tradeoff between the matching and fairness of coverage, and that the pause time parameter provides a means to control this tradeoff for the WRW-aLP algorithm.*

C. Power and Optimal Pause Time

We evaluate the method in IV-A to compute the power of single sensor coverage for Atlanta and Los Angeles.



(a) Atlanta

(b) Los Angeles

Fig. 5. RMSE/unfairness tradeoff by P of WRW-aLP.

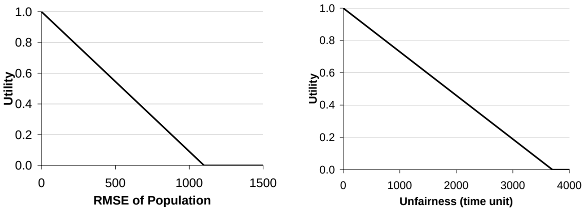
We use the utlitiy functions, $\mathcal{U}_{\mathcal{M}}(\cdot)$ and $\mathcal{U}_{\mathcal{F}}(\cdot)$, shown in Fig. 6 for the RMSE (scaled by the population size) and unfairness, respectively. Because the cities are of different sizes, we use $t_m = 320$ and $t_f = 3500$ time units for Atlanta, and $t_m = 1100$ and $t_f = 3700$ time units for Los Angeles. Fig. 7 plots the power measure achieved against the pause time parameter P, for $\alpha = 0.5, 0.7, 0.9$. Notice that in general, the power increases initially as P increases, because of improved

P (time unit)	1	2	4	8	16	32	64
RMSE average	258.44	201.83	136.35	76.98	37.69	17.30	7.32
25-percentile	258.32	201.57	136.12	76.61	37.60	17.28	7.28
75-percentile	258.76	201.89	136.59	77.21	37.75	17.34	7.35

TABLE II
AVERAGE AND 25-/75-PERCENTILES OF WRW-aLP RMSE OF POPULATION DISTRIBUTION FOR ATLANTA, AS A FUNCTION OF P.

matching. The power reaches a single peak and then decreases afterwards, because a further increase in P causes the unfairness to become too high.

We apply Brent's method (see Section V) to find the optimal P that maximizes the power. For each of the power functions shown for each city, Brent's method converges in less than 12 iterations. For LA and $\alpha = 0.7$, Fig. 8 shows the computed P and the corresponding power achieved, after each iteration of the algorithm. As shown in the figure, the first three iterations are used to bracket the optimal P, and the next seven iterations identify that optimal. Table III summarizes the optimal P computed for the two cities, for $\alpha = 0.5, 0.7, 0.9$. They agree with the highest corresponding power shown in Fig. 7. *We conclude that Brent's method can compute the optimal power parameter accurately and efficiently.*



(a) $\mathcal{U}_{\mathcal{M}}$ (b) $\mathcal{U}_{\mathcal{F}}$

Fig. 6. Utility functions $\mathcal{U}_{\mathcal{M}}$ and $\mathcal{U}_{\mathcal{F}}$.

City	Atlanta			Los Angeles		
α	0.5	0.7	0.9	0.5	0.7	0.9
Optimal P (time unit)	10.00	6.50	2.58	15.95	9.90	4.72
Power	0.70	0.68	0.72	0.75	0.75	0.79
# iterations	12	9	17	9	10	13

TABLE III
OPTIMAL P FOUND BY BRENT'S METHOD FOR ATLANTA AND LA, $\alpha = 0.5, 0.7, 0.9$.

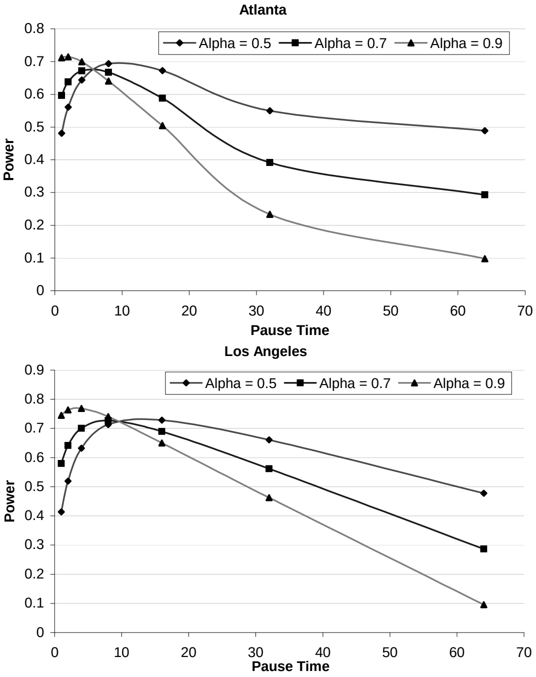


Fig. 7. Power functions.

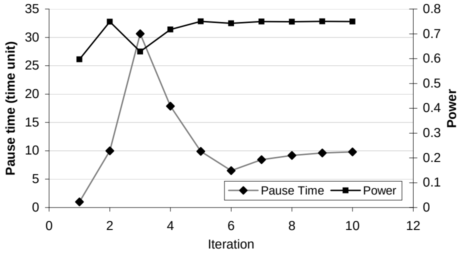
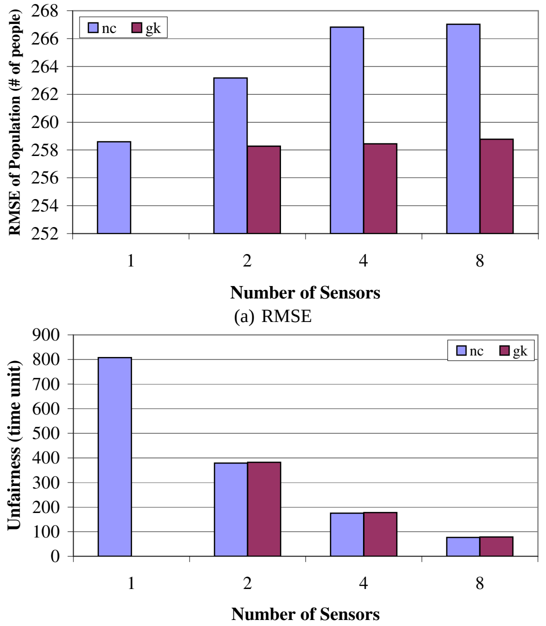


Fig. 8. Computed P and corresponding power achieved after each iteration of Brent's method for LA, $\alpha = 0.7$.

D. Multiple sensors

This set of experiments illustrates the effects of multiple sensors. Figures 9(a) and 9(b) show the unfairness and RMSE of the WRW-aLP algorithm, respectively, for Atlanta. The number of sensors, n, is varied to be 2, 4, and 8. We compare the cases when the sensors operate independenty (the case labeled "nc") or when they have access to the global coverage profile (the case labeled "gk"), as defined in Section VI. Notice from Fig. 9(a) that for both nc and gk, the unfairness roughly halves each time we double the number of sensors, showing that increasing the sensing resources will reap roughly proportionate benefits, for up to 8 sensors and for a large city like Atlanta. However, Fig. 9(b) shows that in contrast to fairness, the steady-state matching does *not* improve as we use more sensors. This is because, for both nc and gk, the global coverage profile of all the sensors will, over the long term, approach the global coverage profile of each individual sensor. Hence, the additional sensors do not fundamentally benefit a long-term performance measure such as matching. Conversely, multiple sensors actually introduce the possibility of inefficieny when more than one sensor visit the same cell at the same time, which may hurt the matching. In the case of up to 8 sensors for Atlanta, the degree of redundant coverage is small. Hence, the RMSE increases slowly.



(a) RMSE

(b) Unfairness

Fig. 9. Performance of WRW-aLP for varying number of sensors in Atlanta.

We further evaluate the impact of the pause time parameter on the RMSE and unfairness results for multiple sensors. The results for Atlanta are shown in Fig 10, for 1, 2, and 4 sensors and both cases of nc and gk. The results show that the nature of tradeoff remains the same in the multiple sensor case as in the single sensor case.

We evaluate the coordination strategies presented in Section VI. We use four sensors for Atlanta. The coordination strategies are as presented in Section VI and include: (1) independent operation (case "nc"), (2) knowledge of global coverage profile (case "gk"), (3) static partitioning by equal accessible area, and (4) static partitioning by equal total threat. For (3) and (4), we implement two different actual partitions that satisfy each of the equal area (cases "ea-1" and "ea-2") and equal threat (cases "et-1" and "et-2") objectives. The achieved unfairness and RMSE of the different coordination approaches are shown in Fig. 11. First, notice that for static partitioning, the performance can be dependent on the actual partition used. The largest difference, though still small, is for the unfairness between ea-1 and ea-2 (134 vs 137 time units). Second, independent operation has highly competitive performance against the coordinated approaches. In fact, it performs the best in all the cases except for fairness under ea-1. This shows that while independent operation can cause redundant coverage, the performance penalty is not larger than the loss of efficiency due to a non-optimal partitioning of the coverage areas, which restricts the ability of one sensor to help monitor a cell assigned to another sensor. Notice that while we have studied only basic coordination approaches, the almost best-case fairness gain with small RMSE for independent operation suggests that, unless the map is small relative to the number of sensors, it will be difficult for any coordination approach to significantly outperform no coordination.

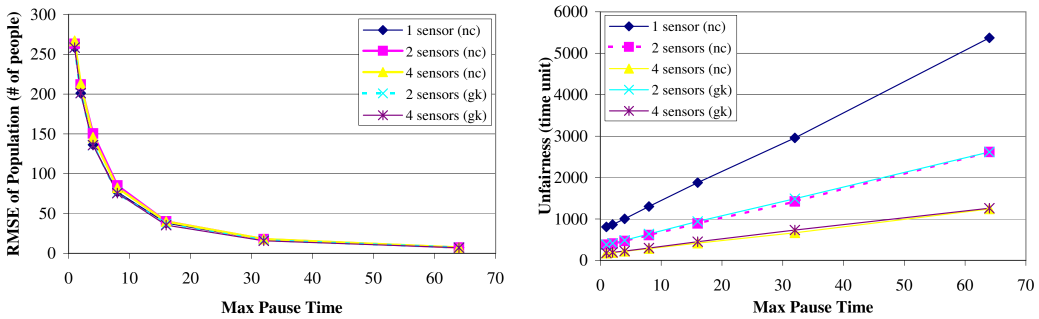
In summary, *we conclude that (1) using more sensors can significantly improve the fairness of coverage (as quantified by a lower unfairness value), although the marginal improvement due to an additional sensor decreases slightly as the number of sensors increases; (2) using more sensors reduces the accuracy of matching slightly, under either independent operation or knowledge of the global coverage profile; and (3) for multiple sensors, basic coordination approaches – based on either knowledge of the global coverage or a static division of responsibilities – do not improve performance over independent operation. The almost best-case performance of n sensors without coordination shows that the stochastic movement enables the benefits of multiple sensors to be largely realized, without additional schedule planning/runtime coordination overheads.* Moreover, Fig. 10 shows that if some of the sensors fail in a stochastic, uncoordinated deployment, the system of sensors will achieve a *graceful degradation* in performance without explicit recovery/replanning actions.

IX. Conclusions

We have formulated the problem of covering a surveillance area by one or more mobile sensors based on a general threat profile. We proposed matching and fairness as basic though antagonistic performance measures of the problem. We showed how a complement of techniques can be combined orthogonally to give a WRW-aLP algorithm that can achieve excellent matching and good fairness at the same time. Moreover, a pause time parameter in WRW-aLP enables a controlled tradeoff between the fairness and matching, and the optimal parameter that maximizes the combined power metric can be efficiently computed.

We showed that the achievable fairness can be limited by the availability of too few sensors for too large an area. In that case, the use of more sensors is effective. Our multiple sensor results, while preliminary, suggest that a simple deployment strategy of independently operating the sensors in stochastic movement, is viable, because it is largely effective while requiring no customized planning based on the number of sensors. Moreover, the independent operation approach degrades gracefully when a subset of the sensors fail, even without explicit recovery/replanning actions.

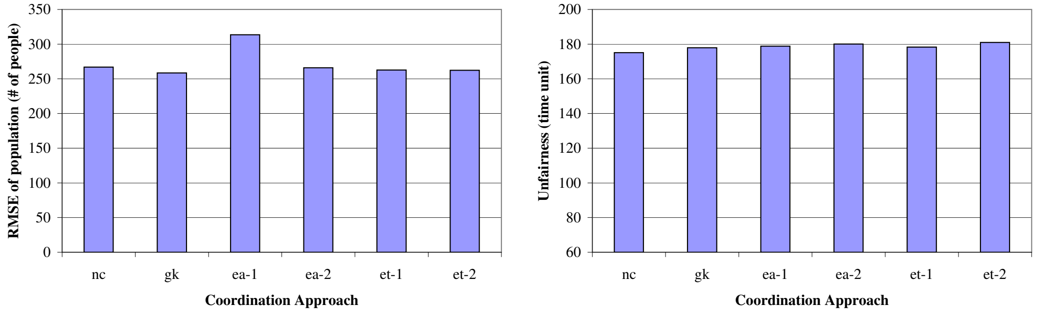
While we assume that the threat profile is static, it is clear that by changing the input of the mobility



(a) RMSE

(b) Unfairness

Fig. 10. RMSE/unfairness tradeoff of WRW-aLP pause time parameter for varying number of sensors in Atlanta.



(a) RMSE

(b) Unfairness

Fig. 11. Performance of WRW-aLP with multiple sensors under different coordination approaches in Atlanta.

algorithm to a new profile, we can eventually adapt to the new threat pattern. Hence, adapting to changes in the threat because of slowly changing conditions – e.g., the distribution of population, the locations of strategically important facilities, and seasonal variations in the weather – should not be a concern. Nevertheless, in the case that the threat profile changes quickly, such as the effects of a sudden storm on plume propagation, the challenge is significantly harder and has not been solved.

We are building a campus-scale sensor testbed based on the proposed mobile coverage algorithms. In the testbed, radiation/chemical sensors are carried by low cost robots that support wireless communication and programmable movement.

References

[1] R. W. Lee and J. J. Kulesz, "A risk-based sensor placement methodology," Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Tech. Rep., 2006.

[2] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *Proc. of IPSN*. Piscataway, NJ, USA: IEEE Press, 2005, p. 55.

[3] X. Du and F. Lin, "Maintaining differentiated coverage in heterogeneous sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 5, no. 4, 2005.

[4] J. Jeong, S. Sharafkandi, and D. H. C. Du, "Energy-aware scheduling with quality of surveillance guarantee in wireless sensor networks," in *Proc. of DIWANS*, 2006.

[5] M. Cardei, "Energy-efficient target coverage in wireless sensor networks," in *Proc. of IEEE Infocom*, 2005.

[6] R. Iyengar, K. Kar, and S. Banerjee, "Low-coordination topologies for redundancy in sensor networks," in *Proc. of MobiHoc*. New York, NY, USA: ACM Press, 2005, pp. 332–342.

[7] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity ofwireless sensor networks," in *Proc. of WICON '05*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 114–121.

[8] X. Bai, S. Kuma, D. Xua, Z. Yun, and T. H. La, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proc. of MobiHoc*. ACM Press, 2006.

[9] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proc. of MobiCom*, 2004.

[10] R. J. Nemzek, J. S. Dreicer, D. C. Torney, and T. T. Warnock, "Distributed sensor networks for detection of mobile radioactive sources," *IEEE Trans. on Nuclear Science*, vol. 51, no. 4, 2004.

[11] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in wireless sensor networks: theory and practical solutions," *Wireless Network*, vol. 8, no. 5, pp. 443–454, 2002.

[12] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *Proc. of IEEE Infocom*, 2004.

[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. of IEEE Infocom*, 2003.

[14] Z. Vincze, D. Vass, R. Vida, A. Vidacs, and A. Telcs, "Adaptive sink mobility in event-driven multi-hop wireless sensor networks," in *Proc. InterSense*, 2006.

[15] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proc. MobiHoc*. New York, NY, USA: ACM Press, 2005, pp. 300–308.

[16] J. P. Hespanha, K. H. Jin, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *Proc. of the 38th IEEE Conference on Decision and Control*, vol. 3, 1999.

[17] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation," in *IEEE Trans. on Robotics and Automation*, 2002.

[18] L. Schenato, S. Oh, and S. Sastry, "Swarm coordination for pursuit evasion games using sensor networks," in *Proc. of the Int. Conf. on Robotics and Automation*, 2005.

[19] N. Bisnik, A. Abouzeid, and V. Isler, "Stochastic event capture

using mobile sensors subject to a quality metric," in *Proc. of MOBICOM)*, September 2006.

[20] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, 1996.

[21] R. Brent, *Algorithms for Minimization without Derivatives*. Prentice-Hall, 1973.

[22] "LandscanTM global population database. oak ridge, TN: Oak ridge national laboratory. available at http://www.ornl.gov/landscan/."