

Detection and Localization Sensor Assignment With Exact and Fuzzy Locations

Hosam Rowaihy¹ Matthew P. Johnson² Diego Pizzocaro³
Amotz Bar-Noy² Lance Kaplan⁴ Thomas La Porta¹ Alun Preece³

¹ Dept. of Computer Science and Engineering, Pennsylvania State University, USA

² Dept. of Computer Science, Graduate Center, City University of New York, USA

³ School of Computer Science, Cardiff University, UK

⁴ U.S. Army Research Laboratory, USA

Abstract. Sensor networks introduce new resource allocation problems in which sensors need to be assigned to the tasks they best help. Such problems have been previously studied in simplified models in which utility from multiple sensors is assumed to combine additively. In this paper we study more complex utility models, focusing on two particular applications: event detection and target localization. We develop distributed algorithms to assign directional sensors of different types to multiple simultaneous tasks using exact location information. We extend our algorithms by introducing the concept of fuzzy location which may be desirable to reduce computational overhead and/or to preserve location privacy. We show that our schemes perform well using both exact or fuzzy location information.

1 Introduction

Mission-centric sensor networks present many research challenges. One such challenge is how to best assign sensors to tasks, considering that there may be multiple tasks, of different priorities and information needs, running concurrently in the network, and sensors of multiple types available to meet those needs. Tasks may require one or more sensors, possibly of different types. Given this multiplicity of task types and needs, our goal is to assign specific sensors to the tasks in order to maximize the utility of the sensor network. This is especially challenging in environments that use *directional* sensors as each sensor in this case can be assigned to at most one task.

In this paper, we consider the problem of assigning directional sensors to tasks. In particular, we focus on two applications, *event detection* and *target localization*. We propose distributed algorithms for assigning specific sensors to tasks of both types. For the two problems we consider a case in which the *exact location* of the sensors is known, and one in which only an approximation of the location is disclosed (we term this *fuzzy location*). Assignment algorithms based on the exact location lead to better solutions and higher overall performance. In certain cases, however, such schemes are not feasible, for two reasons. First, exact location creates a large problem instance in which each sensor is considered on its own, which leads to a higher computational cost. This can be impractical due to the limited computational capabilities of sensors. When fuzzy location is used, however, nearby sensors can be clustered based on their

fuzzy location, thus coarsening the problem instance, and requiring the consideration of fewer assignment choices. Second, exact location may not be disclosed for privacy reasons. Consider a scenario in which a sensor network deployed for a sensitive task (e.g. monitoring national borders). The owners of the network might like to share the sensors with other entities (e.g. researchers collecting environmental data) but at the same time be reluctant to reveal precise information about the location of their assets. By accommodating fuzzy location, we enable such sharing of resources. Different location granularity levels provide trade-offs between performance, and efficiency/privacy.

Contributions. We provide a formal definition of event detection and localization problems. We propose two distributed algorithms, one for the event detection task, and one for the localization task when exact sensor locations are disclosed. The first is guaranteed to provide a 2-approximation; in simulation they both achieve close to optimal performance. We extend the algorithms to cases in which only fuzzy locations of sensors are used. This entails defining the notion of fuzzy location with respect to detection and localization. We show through simulation that, as the granularity of fuzzy location is refined, performance improves to a point after which the gain is insignificant.

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 provides an overview of the problems. Section 4 formally defines the problems. Section 5 introduces our algorithms for assigning sensors to tasks using exact and fuzzy location information. Section 6 shows the performance evaluation results, comparing the different schemes. Finally, Section 7 concludes the paper.

2 Related Work

In the past sensor-task assignment problems in wireless sensor networks have been studied mainly using simplified models in which utility from multiple sensors is assumed to combine additively [?, ?, ?]. [?] uses distributed approaches assign individual sensors to tasks, assuming additive utility and no competition for the same sensing resources between tasks. A problem variant motivated by frugality and conservation of resources is addressed in [?]. In this paper, we consider more complex models to evaluate the utility of a bundle of sensors, and show how such problems can be solved, even based on inexact sensor location information.

Directional sensors with tunable orientations have recently be addressed for coverage [?] and target tracking [?] problems separately. For non-directional sensors, both [?] and [?] propose algorithms to provide a certain level of (cumulative) detection probability *over an area using*. Target localization problems have also been previously considered, e.g. in [?], which develops a solution using a prior distribution of target location and exact sensor locations. Their solution, however, is centralized. A distributed solution for the localization problem is proposed in [?], but it does not consider competition on resources between multiple simultaneous tasks.

Our problem is analogous to the well known Multi-Robot Task Allocation (MRTA) problem described in [?]. A sensor can be seen as a *resource-constrained robot* as suggested in [?], specifically the problem ST-MR-IA of [?], i.e. Single-Task robots (ST) performing Multi-Robot tasks (MR) using Instantaneous Assignment (IA). The MRTA taxonomy solutions, however, do not scale well to large numbers of sensors and tasks.

Our MAXCDP problem (defined below) lies within a family of *submodular* Combinatorial Auctions. Guaranteed approximation algorithms are known for this class of problems (see for example [?] and references therein). Our focus here, however, is on designing algorithms that provide near-optimal performance in an efficient, distributed manner. Some related problems involving cumulative probabilities are considered in [?], but those problems involve the *product* of task success probabilities instead of the sum.

To our knowledge, we are the first to introduce the concept of fuzzy sensor location for sensor-task assignment problems. Related works in this area include [?], which addresses the issue of privacy when fusing data coming from sensors that are assigned to multiple event detection tasks, and [?], which describes a data dissemination technique to ensure that the locations of sensors in the network are not learned by an enemy.

3 Overview

In this section we provide an overview of our network model. Then we discuss the different task types that can be present in the network.

Network Model. The network consists of static sensors of different types. The deployed sensors are *directional* in nature. Examples of such sensors include imaging sensors, which can be used for event detection, and directional acoustic sensor arrays. Thus, we assume that a sensor or a bundle of sensors can be assigned to at most one task at a time. We also assume that sensors know their location.

In our model, a task is specified by a geographic location and a task type, for example, detecting events occurring at location (x, y) or accurately localizing a target within a small area known to contain the target's estimated location. A larger-scale mission, such as field coverage or perimeter monitoring, can be divided into a set of tasks, each having its own location. Because tasks can vary in importance, we allow a sensor to be reassigned from a task with lower profit (which is used to represent importance) to a task with a higher profit. However, since some tasks are more sensitive to interruption in service, preemption should be limited to tasks that can tolerate such interruption. For example, localization is very sensitive to interruption whereas long-term detection is less so.

Task Types. In this paper, we focus on the sensor-task assignment algorithms, and we assume that the process of matching the capabilities of a sensor or a bundle of sensors with the requirements of a task is carried on by a Knowledge Base System [?]. In the network, there may be multiple types of tasks, each having different sensing requirements. Some task types may only require that the assigned sensors are close to the target. Others may require that the collection of sensors form a specific shape, such as in localization.

The specific characteristics of a given task's requirements naturally allow us to restrict our attention to just the applicable sensors. These characteristics are: (1) type of data required, (2) distance from the target, and (3) relative angles between sensors. Together the second and third properties allow the creation of any polygon shape out of the selected sensors to satisfy the requirements of complex tasks.

We consider below two types of tasks incorporating the three requirements. The first task we consider is an *event detection* task in which the goal is to detect activity in a specific location. This task can be accomplished using one or more sensors. Each sensor has a detection probability that depends on its type and distance from the target. A collection of sensors can be combined together to improve the detection probability.

The second task type we consider is a *target localization* task, whose goal is to accurately localize a target within the small area where it is expected, perhaps prompted by the detection of an event in this area or by some prior knowledge. This type of task requires at least two sensors. An interesting property of this task type is that assignment quality depends not only on sensor type and separating distance but also on the angle between the selected sensors. In the model of [?], for example, two sensors perform optimally if they are separated by a 90° angle and are as close to the target as possible.

4 Problem Definition

In this section, we formulate the two sensor-task assignment problems, both of which involve attempts to assign sensor bundles to tasks in the best possible way. In general, we consider the state of the network at an instance of time in which multiple simultaneous tasks can be ongoing. We note that the generic problem of assigning sensor bundles to tasks is a generalized version of the Semi-Matching with Demands (SMD) problem [?], and is NP-hard, even to approximate.

4.1 Event Detection Tasks

The goal of the detection task is to detect events in a specific location with the highest probability. [?] gives a complex model of sensor assignment, with an objective function based on the probability of detecting certain kinds of events, conditioned on the events occurring and the number of sensors assigned to detect the event in a given location. We extract the kernel of this problem as follows. Given are collections of sensors and tasks. Each task is to monitor and detect events, if they occur, in a certain location. The utility of a sensor to a task is the detection probability when the event occurs. Let $S_i \rightarrow T_j$ indicate that sensor i is assigned to task j and let p_j indicates T_j 's profit. The objective function is then to maximize the sum of cumulative detection probabilities for tasks (weighted by task profits), given the probability e_{ij} that a single sensor S_i detects an event for T_j :

$$\sum_j p_j (1 - \prod_{S_i \rightarrow T_j} (1 - e_{ij})) \quad (1)$$

We call this the Cumulative Detection Probability maximization problem (MAXCDP). Here the utilities are monotonic increasing as sensors are assigned but nonlinear.

Proposition 1. MAXCDP is strongly NP-hard.

Proof. We reduce from PRODUCT-PARTITION [?], which is strongly NP-hard (unlike the ordinary PARTITION problem). The input instance is a set of n positive integers

A. The decision problem is to decide whether it is possible to partition them into two subsets $S, S' = A - S$ of equal products, i.e., $\prod_{a_i \in S} a_i = \prod_{a_i \in S'} a_i$. Given the input instance, we produce a MAXCDP instance as follows. There are n sensors and 2 tasks, both with $p_j = 1$. Each sensor S_i has success probability $u_i = 1 - 1/a_i$ for both tasks. Then we have that maximizing:

$$\sum_{j=1,2} p_j (1 - \prod_{S_i \rightarrow T_j} (1 - e_{i,j})) = \sum_{j=1,2} (1 - \prod_{S_i \rightarrow T_j} 1/a_i)$$

is the same as minimizing:

$$\prod_{S_i \rightarrow T_1} 1/a_i + \prod_{S_i \rightarrow T_2} 1/a_i \quad (2)$$

We claim that in an optimal solution, these two products are as close as possible. To see why, consider two values $t_1 = \frac{1}{Ax}$ and $t_2 = \frac{1}{By}$ for the two terms added in Eq. 2, where x, y are particular a'_i values with $A < B$ and $x < y$. Then such a solution must be suboptimal since a “local move” bringing the products closer to equality will strictly reduce their sum:

$$\frac{1}{Ax} + \frac{1}{By} = \frac{Ax + By}{ABxy} > \frac{Ay + Bx}{ABxy} = \frac{1}{Ay} + \frac{1}{Bx}$$

But the terms in Eq. 2 are equal iff we have $\prod_{S_i \rightarrow T_1} a_i = \prod_{S_i \rightarrow T_2} a_i$. Therefore by solving MAXCDP we can decide PRODUCT-PARTITION. \square

We emphasize that the hardness result remains even for geometric instances, indeed, even if sensors and tasks lie on a line. If the detection probability depends on distance then the instances of the reduction can be constructed by placing the two tasks (of different types) at the same point and placing the sensors at distances that yield the desired probabilities.

4.2 Target Localization Tasks

For target localization through triangulation of the bearing measurements, two or more sensors that are not collinear with the target are necessary to ensure full observability of the target’s location. The expected mean squared error when incorporating imperfect bearing measurements is well understood [?, ?]. Specifically, it can be shown that when the bearing measurements are modeled as the true bearings embedded in additive white Gaussian noise (AWGN) of mean zero and variance σ^2 , then the error covariance of the (x, y) location of the target is approximately:

$$\mathbf{R} = \left[\sum_{i=1}^n \frac{1}{\sigma^2 d_i^2} \begin{pmatrix} \cos^2 \theta_i & -\cos \theta_i \sin \theta_i \\ -\cos \theta_i \sin \theta_i & \sin^2 \theta_i \end{pmatrix} \right]^{-1}$$

where d_i and θ_i are the distance and bearing, respectively, from the target event to the i -th sensor. We choose to model the uncertainty in the calculated target location, U , as a function of the expected mean squared error (MSE), which is simply $U = \text{trace}\{\mathbf{R}\}$.

Alternatively, the uncertainty could be $U = \det\{\mathbf{R}\}$ as described in [?]. We prefer the trace because of its physical interpretation as the MSE and because it bounds the determinant.

In this paper, we consider the case in which only two sensors are used for localization, which in most cases provide enough accuracy. More sensors lead to better accuracy but for the purpose of testing our algorithm with exact and fuzzy location two sensors are sufficient. For the case of two sensors, the uncertainty is given by:

$$U = \sigma \frac{\sqrt{d_1^2 + d_2^2}}{|\sin(\theta_1 - \theta_2)|} \quad (3)$$

Note that σ is simply a scaling constant that without loss in generality we ignore by setting to 1. With U thus defined, quality is maximized when the separating angle is 90° and the distances are minimal. Of course, the weights described are just one model. More generally, allowing arbitrary utility weights on triples $t = (s_i, m_j, s_k)$, the following is easily obtained, by reduction from MAXIMUM 3D MATCHING:

Proposition 2. *Choosing a max-weight set of disjoint sensor/task triples is NP-hard.*

5 Distributed Algorithms

In this section we introduce our algorithms for assigning sensors to tasks. We start by discussing the basic operation of the proposed solutions. Then we discuss in detail how to solve the detection and localization problems in the case in which the exact sensor locations are known and the case in which only fuzzy locations are known.

5.1 Basic Operation

The solutions we propose are distributed in nature and do not require any central node to make all the assignment decisions. This allows the leveraging of real-time status information about sensors – which are operational and which are currently assigned to other tasks. Such solutions are also scalable and more efficient in terms of communication cost compared to centralized approaches. We assume a dynamic system, in which the tasks constituting the problem instances described above arrive and depart over time. At any time instance, there may be tasks of the two types present in the network.

For each task, a *leader* is chosen, which is a sensor close to the task’s location. The leader can be found using geographic-based routing techniques [?,?]. If location privacy is a concern, then schemes such as [?] can be used. The task leaders are informed about their tasks’ types, locations and profits by a base station. Each task leader runs a local process to match nearby sensors to the requirements of the task. Since the utility a sensor can provide to a task is limited by a finite sensing range R_s , only nearby sensors are considered. The leader advertises its task information to the nearby sensors (e.g. sensors within two hops). The ad message contains the task type, its location, its profit and location requirement (i.e. exact or fuzzy).

Nearby sensors hearing this ad message will propose to the task with their locations, which may be exact or fuzzy depending on the algorithm used. A sensor assigned to an

Algorithm 1 Exact location algorithm for event detection

initialize each $e'_{ij} = e_{ij}$, the detection probability of S_i for T_i
 initialize each task cumulative detection probability $u_j \leftarrow 0$
 initialize number of assigned sensors to T_j , $n_j \leftarrow 0$
For Task Leader (T_j):
 advertise presence of T_j to each neighboring sensor S_i
 for round = 0 to R **do**
 if $n_j \leq N$ **then**
 among responding sensors G , choose $i \leftarrow \arg \max_i \{e'_{ij} : S_i \in G\}$
 update $u_j \leftarrow u_j + e'_{ij}$
 send accept messages and advertise new u_j
 else done
For Sensor (S_i):
 wait for task requests
 among requesting tasks Q , choose $j \leftarrow \arg \max_j \{e'_{ij}p_j : T_j \in Q\}$
 send proposal to T_j including exact location
 if accepted **then** S_i is assigned to T_j ; **done**
 else
 listen to current u_j values for requesting tasks; **if** no more tasks **then done**
 update detection probability based on new u_j 's: $e'_{ij} \leftarrow 1 - (1 - u_j)(1 - e_{ij}) - u_j$
 repeat

interruptible task may be reassigned to another task if doing so will increase the total profit of the network. This is determined as follows: if the utility sensor S_i provides to the incoming task T_j weighted by T_j 's profit p_j is greater than that of the current task T_k then S_i should be reassigned. More formally, if $e_{ij}p_j > e_{ik}p_k$ then S_i is reassigned. We allow both localization tasks and detection tasks to preempt detection tasks; neither type can preempt a localization task. To reduce both the interruption of ongoing tasks and the communication overhead, no cascading preemption is allowed. That is, if task T_j preempts task T_k , T_k will try to satisfy its demand only with available sensors rather than by preempting a third task.

When a task ends, the leader sends out a message to advertise that the task has ended and all its assigned sensors are released. Because the system is dynamic, tasks that are not satisfied after the first assignment process will try to obtain more sensors once they learn there may be more available. This information can be obtained either from the base station or by overhearing the message announcing the end of a task.

The remaining part of this section describes the algorithms that the task leaders use to determine which sensors should be assigned to each task. As both problems as defined above are NP-hard, our algorithms are heuristic based.

5.2 Exact Location Algorithms

In this subsection we propose algorithms to solve the sensor-task assignment problems, for detection and localization, when the exact locations of sensors are known.

Event Detection Tasks

In order to conserve energy we limit the number of sensors that can be assigned to a

task to N , which is an application parameter. A higher value of N may yield a higher cumulative detection probability for an individual task. Between tasks, however, there will also be greater contention for sensors.

Due to the competition that can occur between tasks we propose an algorithm that runs in rounds to allow sensors to be assigned to their best match. When a task arrives to the network, the task leader advertises the presence of the task and its profit to nearby sensors. The ad message is propagated to ensure that all tasks that are within twice the sensing range ($2R_s$) receive it. Since these tasks compete for the same sensors with the arriving task their leaders need to participate in the process.

In the first round, each leader informs the nearby sensors of the details of its task (location and profit). A sensor, which may hear ad messages from one or more tasks, proposes to its current best match. This is the task for which it provides the highest detection probability weighted by the task's profit. More formally, S_i proposes to task T_j that maximizes $e_{ij}p_j$. From the set of proposing sensors, each task leader selects the sensor with the maximum detection probability and updates its current cumulative detection probability (CDP).

In the next round, each leader sends out an update on the status of its task's CDP after taking into account the currently assigned sensors. Sensors that were not selected in the first round recalculate e'_{ij} , the amount by which they can increase the current CDP of the different remaining tasks (shown in the step before last in Algorithm 1). Again each unassigned sensor proposes to its best fit. This process continues for R rounds until all tasks have N assigned sensors or there are no more sensors available. R is an application parameter and should be set to be equal to at least N to give tasks a chance to assign enough sensors. Algorithm 1 summarizes the steps followed. Note that all the competing leaders will go through the steps shown for the task leader.

During this process sensors can continue the detection for the tasks to which they were initially assigned. Change only happens if a sensor chooses a different task in which case it will be directed towards the new task's location and start detecting events.

When $N = \infty$ and full preemption is allowed, this algorithm provides a 2-approximation guarantee which can be proved by adapting the proof of [?]. It is easy to construct examples showing that the guarantee is tight.

Proposition 3. *Algorithm 1 is a 2-approximation.*

Proof. For a given problem instance I , consider the first time that a sensor proposal is accepted by a task, say T_j accepting S_i , with value $p = v_k(S_i) = p_j e_{ij}$. Let I' be the problem instance after this assignment is chosen and other sensors' proposal values for T_j and its remaining profit (that is, T_j 's valuation $v_j(\cdot)$) are updated appropriately. Suppose S_i is assigned to T_k in optimal solution $OPT(I)$. Then we lower-bound the value of $OPT(I')$. One possible solution is the same as $OPT(I)$ except with neither T_j nor T_k receiving S_i . T_j 's potential profit is reduced by exactly p . Since S_i greedily proposed to T_j rather than to T_k , in the solution to instance I' we must have profit from T_k reduced by at most p . Thus $OPT(I) \leq OPT(I') + 2p$. By applying the argument inductively, we obtain the result. \square

Target Localization Tasks

We propose a simple distributed solution (Algorithm 2) to the exact location localiza-

Algorithm 2 Exact location algorithm for target localization**For Task Leader (T_j):**

advertise presence of T_j
 receive sensor proposals
 among responding sensors G
 choose $(i, k) \leftarrow \arg \min_{i,k} \{(\frac{\sqrt{d_1^2 + d_2^2}}{|\sin \theta|}) : (S_i, S_k) \in G^2\}$
 send accept messages

For Sensor (S_i):

receive task request
if (S_i not assigned to localization task) **then** send proposal to T_j including the exact location
else ignore request
if accepted **then** S_i is assigned to T_j ; **done**

tion problem. The goal in localization is to minimize the achieved uncertainty of the assigned sensor pair. Because localization tasks are sensitive to preemption, only nearby sensors that are *not assigned* to any other localization task propose to the leader with their exact location. If a sensor is assigned to a task that is less sensitive to preemption, such as detection in our case, it will also propose to the task. Among the proposing sensors, the leader chooses the pair of sensors that provides the lowest uncertainty according to Eq. 3. To do this each sensor must provide its angle from a predetermined axis. The angle of a sensor can be measured from the y-axis that passes at the estimated target position. For a pair of sensors, the separating angle (θ) can then be determined by calculating the absolute difference between their respective sensor angles.

A task's number of neighboring sensors (of the needed type) will typically be limited and so considering all sensor pairs should be feasible. If there are many proposing sensors, the leader can set a distance threshold and ignore any sensors beyond this point. After making the assignment decision, the leader sends messages to the selected sensors. If they were previously assigned to other tasks, the leaders of those tasks are informed that they should search for replacements.

5.3 Fuzzy Location Algorithms

In the previous subsection we proposed algorithms to assign sensors to tasks based on their exact locations. However, in some situations these schemes might not be feasible, either due to computational cost or due to location privacy concerns. In this subsection, we propose algorithms to assign sensors based only on their fuzzy locations. Instead of having the assignment algorithms to consider each sensor on its own, fuzzy location allows sensors to be classified into classes based on their fuzzy location. We consider the distance and angle requirements introduced in Section 3 to make the assignment based on different granularities. Recall that in this case sensors know their exact locations but do not disclose them.

Event Detection - Fuzzy Distance

In event detection, the probability that a sensor detects an event depends heavily on

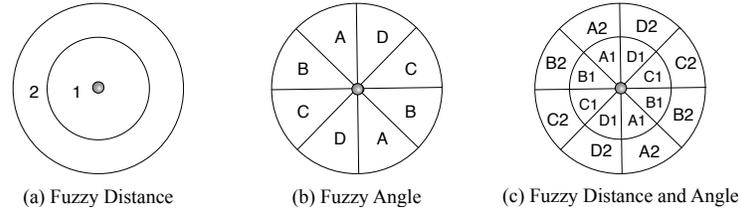


Fig. 1. Fuzzy Location

the distance between them. Here we define *fuzzy distance* based on different *distance granularities* as a measure of a sensor's location. Clearly, only sensors within sensing range a task (location) should be considered. This area can be represented as a circle with radius R_s centered at the task location. If no distance granularity (DG) is specified (i.e. $DG = 0$), then all sensors within this circle are considered to be in the same class and equivalent. A solution based on $DG = 0$ will provide almost no guarantee on the solution quality. When DG is increased to 1, the distance from the target to the edge of the circle is divided to create two rings or *annuli* of equal area, which partitions the sensors into two classes. In Fig. 1(a) we see an example of fuzzy distance based on $DG = 1$. A sensor of class 1 will provide higher detection probability than a sensor of class 2. $DG = 2$ divides the circle into three equal-area rings, and so on.

The algorithm used for detection is similar to Algorithm 1 above, with the change that sensors report back their classes rather than their exact detection probability. After the task leader sends out the task advertisement message, nearby sensors hear the message and classify themselves based on the distance granularity specified in the leader's message. If a sensor is currently assigned to another task, it can decide, based on the preemption rules discussed above, to propose to the new task. The leader then chooses the best sensors for its task which in this case are the ones that lie within the closest rings. The detection probability of a sensor is determined based on the expected distance from a point in the ring in which the sensor lies to the center of the circle.

This process not only provides location privacy by hiding the exact location of sensors but also reduces the computation time required to choose the assignments. The leader needs to consider only $DG + 1$ classes of sensors, instead of all individual sensors. Clearly, the higher the DG value, the better the selection becomes, which leads to a higher cumulative detection probability. On the other hand, a larger DG yields more sensor classes and sensor location privacy. The granularity level is a system parameter which we study below. We note that even if the exact distance from the sensor to the target is known, the task leader cannot accurately locate the sensor since it can be anywhere around a circle. Therefore, fuzzy distance is more protective of sensor privacy than is fuzzy angle, which we consider next.

Target Localization - Fuzzy Angle

To accurately localize a target, the task leader should not only pick sensors close to the target but also sensors with a separating angle as close as possible to 90° . This suggests another form of fuzzy location, based on the sensor's viewing angle.

To use fuzzy location, a sensor needs to determine its *fuzzy angle*. This is done based on the *angle granularity* (AG), which is indicated by the sector angle (given a circle centered at the estimated target location with radius R_s). For example, when $AG = 360^\circ$, all sensors within the circle are placed in the same class regardless of angle. If $AG = 90^\circ$, then the circle is partitioned into four quadrants and the sensors are partitioned into four classes. When a sensor hears a task ad message, it determines its actual angle (using the y-axis that passes in the target estimated location as a reference) which then determines its sector. Note that since we only need to calculate $|\sin \theta|$, where θ is the separating angle, to determine the uncertainty of a sensor pair, sensors in opposite sectors are considered to be in the same class. Fig. 1(b) shows a circle divided into eight 45° sectors, i.e. $AG = 45^\circ$.

Since the target localization uncertainty model depends on both the separating angle and the distance, the fuzzy location comprises both the fuzzy distance and fuzzy angle. After dividing the circle into sectors, we divide it into rings based on the distance granularity (see Fig. 1(c)). The algorithm used for localization in this case is similar to Algorithm 2 above. The difference is a sensor’s proposal now indicates its sector rather than its precise location. The leader runs the algorithm on all sensor classes (using the expected distance and expected angle for each) to determine the best pair of classes using Eq. 3. From each class a sensor is chosen arbitrarily. Note that with finer granularities, some sectors might be empty and hence some classes can be ignored.

The number of sensor classes in this case is a function of both DG and AG . Assuming for simplicity that in each increment of granularity we divide AG by two, then the number of classes becomes $(DG + 1)(180/AG)$. Note the special case when $AG = 360^\circ$ in which we will have one class. Also, note that $AG = 180^\circ$ is not used in our experiments as sensors from the two sectors will be equivalent. As with fuzzy distance, finer location granularity leads to more sensor classes and hence higher computational overhead. Also, with finer granularity the task leader gains more information about a sensor’s location, which decreases privacy.

6 Performance Evaluation

In this section we discuss experiments evaluating our algorithms. We implemented a simulator in Java and tested our algorithms on randomly generated problem instances. We compare the results achieved by both the exact and fuzzy location algorithms. We also study the effect on detection quality of the maximum number of sensors a detection task allowed to receive. Finally, we analyze the effect of the location granularity level on the computational overhead and the sensors’ location privacy.

6.1 Simulation Setup

There are two types of deployed sensors, directional acoustic sensors and imaging sensors, and two task types, detection and localization. The localization task can only utilize acoustic sensors, which must be assigned in pairs. Detection tasks can utilize both sensor types but to varying effect. Detection means that the beamformed output yields evidence of a target at a given bearing direction. The sensors need not be positioned to

provide precise triangulation of the target. On the other hand, localization requires the sensors to be positioned so that the triangulation error for the target location is within given bounds as dictated by the utility function. The target location uncertainty of a pair of sensors to a task is found using Eq. 3.

The detection probability with sensor S_i assigned to task T_j is defined as follows:

$$e_{ij} = \exp\left(\log(P_{FA})\left(1 + \frac{SNR_1}{D_{ij}^2}\right)^{-1}\right) \quad (4)$$

where D_{ij} is the distance between the sensor and the task location, P_{FA} is the false alarm probability (a user-chosen parameter), and SNR_1 is the normalized signal-to-noise ratio at a distance of one meter from the source signal. This expression results from analyzing a fluctuating source model embedded in AWGN when the square law detector is employed [?]. For computational and analytic convenience, we simply approximate e_{ij} as zero when D_{ij} exceeds an effective sensing range of the sensor $R_s = 40m$. SNR_1 was set to $60dB$ for acoustic sensors and to $66dB$ for imaging sensors. (Imaging sensors are assumed to have higher SNR due to their higher fidelity and zooming capabilities.) For both types, we set $P_{FA} = 0.001$. These functions are only used for testing in our experiments and are not properties of our schemes; they are not meant to model the exact behavior of these two types of sensors. In our experiments, 30% of the sensors are imaging and 70% acoustic.

Our goal is to maximize the achieved profits from all available tasks, i.e. $\max \sum_j p_j u_j$ where u_j the the utility received by task T_j and p_j is its profit. The utility achieved by a detection task is the cumulative detection probability (CDP), which is naturally in $[0,1]$. The utility that a pair of acoustic sensors provide to a localization task depends on the uncertainty level (Eq. 3). We normalize this value to $[0,1]$ by treating acceptable uncertainty value (an application-specific parameter) as full utility. In our experiments we set this value to 16, which represents an error area of 4m in width. Any selected pair with uncertainty under 16 has 100% utility. Higher uncertainty means less utility; for example, uncertainty of 64 indicates 25% utility.

We deploy 1000 nodes in uniformly random locations in a $400m \times 400m$ field. The communication range of sensors is set to 40m. Tasks are created in uniformly random locations in the field. Localization tasks profits vary uniformly in $[0.1,1]$; detection task profits vary uniformly in $[0,0.1]$, on average, an order of magnitude lower. We assume that these profits are awarded per unit of time for which a task is active. The maximum possible profit in time step is the sum of profits of all active tasks at that time step.

Task lifetimes are uniformly distributed. Detection tasks, by their nature, last much longer than localization tasks, which are discrete computations typically prompted by particular detected events. Localization task lifetimes vary uniformly between 5 and 30 minutes, whereas detection task lifetimes vary uniformly between 1 and 5 hours. Tasks arrive based on a Poisson process, with an average arrival rate of 10 tasks/hour. Mirroring the sensor distribution, 30% of tasks are for localization and 70% are for detection.

To test our algorithms, we compare their performance with an upper bound on the optimal solution quality. For each currently active task *separately*, we find opti-

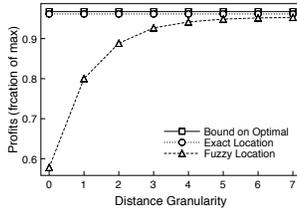


Fig. 2. Detection

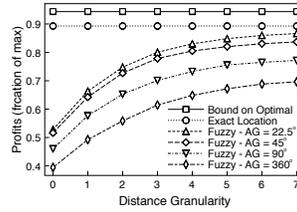


Fig. 3. Localization

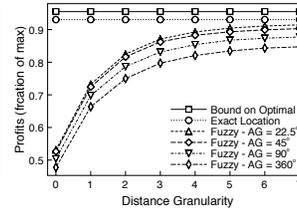


Fig. 4. Overall Performance

mal achievable profit for it, assuming there are no other tasks in the network, i.e. no competition. The sum of these values provides a (loose) upper bound.

In our experiments, we show the average performance of the network for a period of 50 hours; we take the measurements at steady state after running the algorithms for 10 hours. Each point in the graph represents the average achieved profit per unit of time as a fraction of the maximum possible profit. The results are averaged over 20 runs.

6.2 Simulation Results

Fig. 2 shows the average performance of the detection tasks. We limit the number of sensors that a task can have to 5 (i.e. $N = 5$). For Algorithm 1 we set the number of rounds $R = N$. We compare the results achieved by the exact and fuzzy location algorithms. We vary the distance granularity (DG) from 0 to 7 and observe its effect on the fuzzy location performance. The achieved profits initially increase rapidly as DG increases, but then slow once DG reaches 4. This suggests that the benefit gained from the increased granularity may not justify the loss in privacy and the increase in the computation cost. By the time DG reaches 7, the fuzzy location scheme performance is within less than 1% of the exact location scheme, which itself is near-optimal.

Fig. 3 shows corresponding results for the localization tasks. We vary both DG and the angle granularity (AG). When $AG = 360^\circ$, i.e. when all sensors within range are placed in the same class regardless of angle, the performance is lowest, as expected. Achieved profits increase along with AG but this increase becomes negligible (less than 1%) once AG becomes finer than 22.5° . We note that the performance of the exact location scheme is within 6% of the optimal bound which is worse than the case of detection. This is mainly due to contention between tasks for the same sensing resources; localization is more sensitive to choice of sensors than is detection, since both distance and angle matter. Combining the results of both schemes (Fig. 4), we find that the total network profits are affected by both previous results.

To study the communication overhead of our algorithms we assumed perfect communication channels. We found that the average number of messages exchanged for a detection task using exact location is 114. For fuzzy location we see a saving of about 10% in the number of exchanged messages due to the clustering of sensors which makes the algorithm converge faster. The algorithm for target localization requires one round (with possible reassignments) and uses only one type of sensors. Hence, its communication overhead is lower; the average number of messages exchanged is 62 messages/task.

We also studied the effect of varying the task arrival rate. Cutting the arrival rate to 5 tasks/hour (and using the previous ratio of task types) leads to a 3% increase in profits

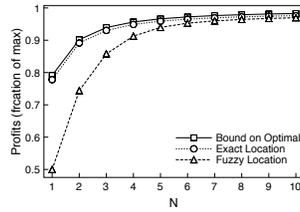
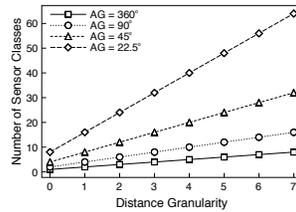
Fig. 5. Effect of Varying N 

Fig. 6. Computational Cost

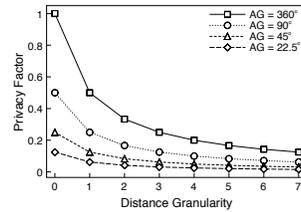


Fig. 7. Privacy

for localization tasks using both exact and fuzzy location. This is attributed to lower competition. Detection tasks do not face considerable competition when the arrival rate is set to 10 tasks/hour and hence we do not see any change in performance when the arrival rate is lowered. Doubling the arrival rate to 20 tasks/hour increases the competition which leads to lower profits for both task types. Profits from detection tasks decrease by a negligible 1% whereas localization tasks witness a 4% decrease. Localization is affected more by changing the arrival rate as it is more sensitive to sensor choice.

In Fig. 5, the performance of the detection algorithms is measured as N increases from 1 to 10. We use an arrival rate of 10 tasks/hour and fix $DG = 5$ and $AG = 2$. Note that a higher value of N means that more sensors can be assigned to each detection task, which will increase CDP. As expected, the profits of the detection tasks increase along with N . The behavior is similar for the exact solution and the upper bound on the optimal solution. The increase is rapid in the beginning but slows down due to the submodular nature of CDP.

6.3 Analysis of Computational Overhead and Privacy

To analyze the computational overhead we plot (in Fig. 6) the number of sensor classes as the granularity of angle and distance becomes finer. When we increase the granularity the number of classes increases as well. The tradeoff between performance and efficiency depends on the number of sensors within sensing range of the task. In our experiments there are on average 31 sensors in that range. For a localization task, if we were to use $DG = 3$ and $AG = 22.5^\circ$, we will end up with 32 classes which is greater than the expected number of sensors surrounding the task. For lower granularities, however, fuzzy location can lower computation cost. Also, in many cases the generated classes will be empty, meaning that fewer classes need be considered. Note that for tasks that only depend on distance, such as detection tasks, the computational savings can be significant.

Fig. 7 shows a privacy factor (P_F) for the different fuzzy location granularities. P_F is the inverse of the number of classes in the previous figure. Below, we use this factor to define two notions of privacy: (1) sensor's anonymity and (2) sensor's location privacy.

For anonymity we use the notion of k -anonymity [?]. Our system provides k -anonymity if the proposal from one sensor cannot be distinguished from at least $k - 1$ other sensors. Let N_s be the number of nodes within sensing range of a mission. Since we divide the circle into equally-sized sectors and assume uniformly random distribution of sensors, if we multiply P_F by N_s we get the average number of sensors lying in a sector. Hence, in our settings $k = P_F \times N_s$. If $P_F = 1$, then a proposing sensor

could be any of the N_s sensors, which provides the highest anonymity. On the other hand, if $P_F \leq 1/N_s$ then $k \leq 1$ which leads to almost no anonymity. Note that the anonymity level is affected by the network density. The more sensors deployed, the higher the value of N_S and hence the better the anonymity.

Sensor's location privacy is defined as the size of the sector in which a sensor lies. If R_s is the sensing range of a sensor, then $P_F \times R_s$ is the area of a sector given a certain fuzzy location granularity. Clearly, the larger the area the better the location privacy since a sensor can be located anywhere within that area. Location privacy is affected by the sensing range. Given that everything else is equal, the larger the sensing range the larger the sector area will be which leads to better location privacy.

We see from Fig. 7 that although P_F stays relatively high when only distance granularity is increased, it decreases rapidly as we divide the circle surrounding the mission location more finely. This adversely affects both the anonymity and location privacy.

7 Concluding Remarks

Although in this paper we limited sensors to performing one task at a time, this limitation is not applicable to all domains. For some sensing data types, e.g. ambient temperature, a sensor may be able to serve many tasks at once. In a sensor network, there may, in fact, be sensors of both types. In this paper, however, we focused on the restricted type of sensor such as directional sensors *since it is the more difficult problem*. In future work, we will consider settings in which sensors of both types are present.

In terms of location privacy, we note that with repeated requests by tasks in the surrounding area of a sensor, an entity can gain more precise information about the sensor's location. This can be learned by considering the intersections of the circles with radius R_s around each task's location. We intend to study such issues in the future.

Acknowledgment This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. N. Ahmed, S. S. Kanhere, and S. Jha. Probabilistic coverage in wireless sensor networks. In *LCN 2005*, Washington, DC, USA.
2. J. Ai and A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1):21–41, Feb. 2006.
3. A. Bar-Noy, T. Brown, M. P. Johnson, T. La Porta, O. Liu, and H. Rowaihy. Assigning sensors to missions with demands. In *ALGOSENSORS 2007*.
4. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. 1999.

5. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.
6. Y. Cai, W. Lou, M. Li, and X. Li. Target-Oriented scheduling in directional sensor networks. In *INFOCOM 2007*, 2007.
7. D. Fotakis and P. G. Spirakis. Minimum congestion redundant assignments to tolerate random faults. *Algorithmica*, 32(3):396–422, 2002.
8. C. Frank and K. Omer. Algorithms for generic role assignment in wireless sensor networks. In *SenSys 2005*.
9. B. P. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in Multi-Robot systems. *The International Journal of Robotics Research*, 23(9):939, 2004.
10. M. Hefeeda and H. Ahmadi. A probabilistic coverage protocol for wireless sensor networks. *ICNP 2007*, pages 41–50.
11. M. P. Johnson, H. Rowaihy, D. Pizzocaro, A. Bar-Noy, S. Chalmers, T. La Porta, and A. Preece. Frugal sensor assignment. In *DCOSS 2008*.
12. I. Kadar. Optimum geometry selection for sensor fusion. In *SPIE 1998*.
13. L. Kaplan. Local node selection for localization in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):136–146, January 2006.
14. L. M. Kaplan and Q. Le. On exploiting propagation delays for passive target localization using bearings-only measurements. *J. of the Franklin Institute*, 342(2):193–211, Mar. 2005.
15. B. Karp and H. Kung. Greedy perimeter stateless routing for wireless networks. In *MOBI-COM 2000*.
16. A. Kelly. Precision dilution in triangulation-based mobile robot position estimation. In *Proceedings of Intelligent Autonomous Systems*, Amsterdam, 2003.
17. C.-C. Lam, P. Sadayappan, and R. Wenger. Optimal reordering and mapping of a class of nested-loops for parallel execution. In *LCPC 1996*.
18. B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *EC 2001*.
19. K. H. Low, W. K. Leow, and M. H. A. Jr. Autonomic mobile sensor network with self-coordinated task allocation and execution. *IEEE Trans. on Systems, Man and Cybernetics (C)*, 36(3):315–327, 2006.
20. K. Mehta, D. Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *ICNP 2007*.
21. A. Preece, M. Gomez, G. de Mel, W. Vasconcelos, D. Sleeman, S. Colley, G. Pearson, T. Pham, and T. La Porta. Matching sensors to missions using a knowledge-based approach. In *SPIE DSS 2008*.
22. A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *MOBICOM 2003*.
23. M. Roughan and J. Arnold. Multiple target localisation in sensor networks with location privacy. In *ESAS 2007*.
24. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, October 2002.
25. S. J. Tutton. Optimizing the allocation of sensor assets for the unit of action. Technical report, Naval Postgraduate School, California, 2006.
26. H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization. In *IPSN '04*, Berkeley, California, USA, 2004.