Exact and Fuzzy Sensor-Task Assignment

Hosam Rowaihy*, Matthew P. Johnson§, Diego Pizzocaro‡,
Amotz Bar-Noy§, Lance Kaplan†, Thomas La Porta* and Alun Preece‡
*Department of Computer Science and Engineering, Pennsylvania State University, USA
§Department of Computer Science, City University of New York, USA
‡School of Computer Science, Cardiff University, UK
†U.S. Army Research Laboratory

***Abstract*—Sensor networks introduce new resource allocation problems in which sensors need to be assigned to the tasks they best help. In the past, such problems have been 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 sensing resources of different types to multiple simultaneous tasks that have different information needs. We show that our schemes perform well using both exact location information or fuzzy location information, which may be desirable to save on computational overhead and/or for privacy reasons.**

I. Introduction

Mission-centric sensor networks present many research challenges. One such challenge is how to best assign sensors to missions, considering that there may be multiple missions, of different priorities and information needs, running concurrently in the network, and sensors of multiple types available to meet those needs.

A mission comprises a set of tasks, each of which requires one or more sensors, possibly of different types. In addition, there may be more than a single combination of sensor types that will satisfy a task. We refer to a combination of sensors as a *sensor bundle*. Likewise, the number of sensors required to satisfy a task may vary depending on their deployment. 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.

Specifically, we consider *event detection* and *target localization* tasks. The detection tasks can be accommodated using any type of sensor that detects activity; here we model acoustic and imaging sensors. The localization tasks can be accommodated using a pair of acoustic sensors. We propose a distributed solution for assigning specific sensor instances. This allows multiple entities to use the sensor network without coordination, which is an important asset for operations that include different allies in a coalition, for example.

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*). Exact location assignment schemes 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 larger problem instance in which each sensor is considered on its own, which leads to a higher computation time. 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 coalition of entities deploy sensors in a field. The various entities might like to share sensing resources but at the same time be reluctant to reveal to one another too much or too precise information about their assets. Different granularity levels provide trade-offs between performance and privacy. By accommodating fuzzy location, we enable coalition partners to share resources without fully disclosing the details of their respective deployments.

Our main contributions and findings are:

- We provide a formal definition of the abstract problem of assigning bundles of sensor instances to tasks, as well as specific problem definitions for event detection and localization tasks. We show that both of these problems are NP-hard.
- We propose two distributed schemes, one for the event detection task, and one for the localization task when exact sensor locations are disclosed. Through simulation we show that they achieve close to optimal performance.
- We extend the schemes 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 II provides an overview of the problems. Section III formally defines the problems. Section IV introduces our algorithms for assigning sensors to tasks using exact and fuzzy location information. Section V shows the performance evaluation results, comparing the different schemes. Section VI discusses related work. Finally, Section VII concludes the paper.

II. Overview

In this section we provide an overview of our network model. Then we discuss the process to determine what sensor or bundle of sensors are required by a task. Finally, we discuss the different task types that can be present in the network.

A. 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. We assume a dynamic system in which tasks arrive and depart over time. When a task arrives in the system, sensing resources are assigned to it. 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.

B. Sensor Bundles

To determine the types of sensors needed and the way they need to be bundled to satisfy the information needs of a task, we use a two-step process. First, we use a Knowledge-Based System (KBS), such as the one described in [1], to determine the combinations of sensor types that may be used to satisfy the task. We refer to these groupings as *bundle types*[1] (BT).

A BT is a set of constraints defining the structure of a bundle, including the types of sensor a bundle should contain, and cardinalities. For example: exactly 2 *acoustic* sensors or at least one *acoustic* sensor and one *imaging* sensor. To generate these BTs, the KBS uses explicit representations of various types of sensors and tasks, in the form of *sensor and task ontologies* [2]. The KBS is given a set of tasks of known types and a set of available sensor types; it uses a semantic matchmaking process [3] to identify combinations of sensor types that can meet the information requirements of each task — these combinations of sensor types yield the BTs.

Given the output of the KBS, the goal of the second step is to determine the best bundle-task assignment that matches the bundle type description to maximize the benefit of the sensor network. This is achieved by the *sensor-task assignment algorithms*. For each (*task type*, *bundle type*) pair, an appropriate computational model from the knowledge base is used to choose the exact sensor instances required for each sensor type within the bundle, and finally evaluate their joint utility. The joint utility is based on the features of the sensor instances deployed in the task's proximity (such as distance from the target or angle from a pre-specified axis). The sensor-task assignment algorithms are the focus of this paper.

Note that we assign a single bundle to each task because a bundle will satisfy all the information requirements of a task by construction. Also, we ensure that each individual sensor is included in at most one bundle since we assume that a sensor can serve only one task at a time. Together, the two steps gradually reduce the search space and thus the convergence time for our sensor-task assignment process, by first restricting assignments to sensors of appropriate types, and then making assignment decisions with regard to individual sensors.

C. Task Types

There may be multiple types of tasks present in the network simultaneously, with various characteristics. In many scenarios, a task will require more than one sensor in order to satisfy its requirements. For example, a larger number of sensors given to a detection task will increase the detection probability, but even a single sensor may provide some help. For other tasks, such as localization, there may be a minimum number of sensors required to obtain any benefit.

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 polygon shape, such as in triangulation or localization. The bundle type requirements of most tasks can be modeled using specific characteristics that limit the number of sensors considered by restricting our attention only to those applicable to the given task. 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. Usually such tasks can use any sensor that can detect activity, e.g. acoustic, imagery, seismic, etc.

The second task type we consider is a *target localization* task. In this task, the goal is to accurately localize a target within a small area in which it is expected to appear, 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. In the model of [4], two acoustic sensors perform optimally if they are separated by a $90°$ angle and as close to the target as possible. 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. For a more accurate localization, more sensors can be used with different separating angles.

III. Problem Definition

In this section, we formulate a generic *Bundle-Task Assignment Problem*, and then two special cases of it, all of

[1] In [1] they are called "package configurations".

which involve attempts to assign sensor bundles to tasks in the "best" possible way. For simplicity, we consider the state of the network at an instance of time in which multiple simultaneous tasks can be ongoing.

A. Abstract Problem

The Bundle-Task Assignment Problem can be modeled as a tripartite graph whose vertices consist of a set of sensors $S = \{S_1, \ldots, S_n\}$, a set of tasks $T = \{T_1, \ldots, T_m\}$ and a set of bundles $B = \{B_1, \ldots, B_l\}$.

For each task, we are given a set of sensor bundles, each of which would at least minimally satisfy the task. Each possible assignment of a bundle k to a task j is associated with a profit value p_j that the task will thereby achieve, which is based on the task's inherent importance, and the amount of utility e_{kj} it receives. The goal is to maximize the sum of the utilities for the tasks (weighted by their profits), subject to the constraint that no sensor or task is used more than once. This problem is essentially the NP-hard SET PACKING problem [5], and it can formulated as an integer program (IP) as follows:

$$\begin{array}{ll} \textit{max:} & \sum_{kj} p_j e_{kj} \cdot y_{kj} \\ \textit{s.t.:} & \sum_{k} y_{kj} \leq 1, \text{ for all } j, \\ & \sum_{kj} I_{ik} y_{kj} \leq 1, \text{ for all } i \\ & y_{kj} \in \{0,1\} \text{ for all } j,k \end{array}$$

We now explain the IP. The decision variables y_{kj} indicate whether bundle k was assigned to task j. The first set of constraints prevents more than one bundle from being assigned to any one task. Matrix I specifies the membership relationship between sensors and bundles. The second set of constraints prevents any sensor from being used more than once, in multiple chosen bundles.

For small instances, optimal solutions can be obtained by solving this IP. As a generalization of the Semi-Matching with Demands (SMD) [6], however, the problem is NP-hard, even to approximate. Larger problem instances of the generic problem may be solved by heuristics.

Our main interest here, however, is in two particular special cases of the abstract problem, involving event detection and target localization. If we know in advance that all the tasks considered in the problem instance are going to be of only a single type, then we can use this information to refine the formulation of the problem. This helps us to develop more specific solutions as we describe in Section IV, although these special cases remain NP-hard. We discuss these problems formally in the subsections that follow.

B. Event Detection Tasks

In [7], a complicated model of sensor assignment is given, 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 probability that it will successfully detect the event if it occurs. Let $S_i \rightarrow T_j$ indicate that sensor i is assigned to task j. 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:

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

We call this the Cumulative Detection Probability maximization problem (MAXCDP). Here the utilities are monotonic (each sensor potentially raises the detection probability further) but nonlinear. Implicitly, this model treats the `OR` of the n individual detection events as a positive detection; alternatively, these events could be `AND`ed together, or more generally we could require q of n detection events be positive for detection. We find that already the `OR`-based problem, MAXCDP, is NP-hard (see [8]).

C. 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 [9], [10]. 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 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 [4]. 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. For the case of two sensors, the uncertainty is given by:

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

Note that σ is simply a scaling constant that without loss of generality we ignore by setting it to a value of one.

For this definition of U, the quality will be maximized when θ is 90° and the distances are as small as possible. Overall, the problem of determining the best pair of sensors is NP-hard (see [8]).

IV. DISTRIBUTED ASSIGNMENT ALGORITHMS

In this section we introduce our algorithms for assigning sensors to tasks. We discuss how to solve the detection and localization problems when the exact sensor locations are known and when only fuzzy locations are known.

A. 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 a scheme that runs in rounds to allow sensors to be assigned to their best match. When a task arrives to the network, the task leader (which is a node close the to location of the task) announces the presence of the task and its profit to nearby sensors. The announcement message is propagated to ensure that all tasks that are within twice the sensing range 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 announcement 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 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.

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.

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

Target Localization Tasks
We propose a simple distributed solution to the exact location localization problem. The goal in the localization task 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. 2.

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. Table 2 shows the steps taken by the task leader and nearby sensors.

B. 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 privacy concerns. In this section, 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 bases on their fuzzy location. We consider the distance and angle requirements introduced in Section II-C to make the assignment based on different granularities.

Event Detection - Fuzzy Distance
In event detection, the probability that a sensor detects an event depends heavily on the distance between them. So, here we define *fuzzy distance* based on different *distance granularities* as a measure of a sensor's location. Clearly, only sensors that are within the sensing range from the task's 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

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

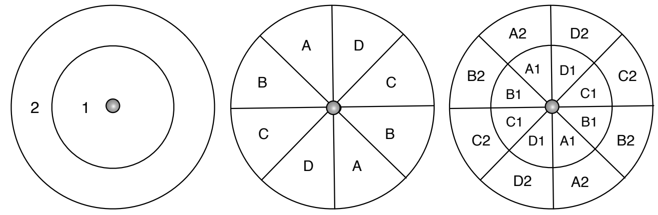
For Sensor (S_i):
receive task request
if (S_i *not assigned* to localization task) **then**
send proposal to T_j including the exact location
else ignore request
if accepted **then** S_i is assigned to T_j; **done**

(DG) is specified (DG = 0) then all sensors within this circle are considered equal (i.e. in the same class). 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 areas. This 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 rings of equal sizes 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 announcement message, nearby sensors hear the message and classify themselves into different classes 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 rules discussed above, to offer itself to the new task. After that, it replies back to the leader with its class. 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 privacy but also reduces the computation time required to choose the assignments. The leader needs to consider only $DG + 1$ classes of sensors, instead of individual ones. Clearly, the higher the value of DG is, the better the selection becomes, which leads to a higher cumulative detection probability. The tradeoff is that this also leads to having more sensor classes and requires providing more precise location information that may compromise privacy. The decision on the level of fineness in the granularity is a system parameter which we study below. We find in our experiments that although an increase in fineness leads to better detection probability, the difference between two granularity levels becomes negligible at some point.

We note that even if the exact distance from the sensor to the target is known, that task leader cannot accurately locate



(a) Fuzzy Distance (b) Fuzzy Angle (c) Fuzzy Distance and Angle

Fig. 1. Fuzzy Location

the sensor since it can be anywhere around a circle. Therefore, fuzzy distance is less susceptible to divulging the location of a sensor compared to the angle, which we consider next.

Target Localization - Fuzzy Angle

To accurately localize a target, the task leader should not only pick sensors that are close to the target but also sensors that have a separating degree that is as close as possible to 90°. This suggests another form of fuzzy location which is based on the angle from which the sensors view the target. The angle of a sensor can be measured from the *y*-axis that passes at the estimated target position. For two sensors, the separating angle (θ) can then be determined by calculating the absolute difference between their respective sensor angles.

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°, all sensors within circle are placed in the same class regardless of angle. If AG = 90°, then the circle is partitioned into four quadrants, each of whose sensors are placed in the same class. When a sensor hears a task advertisement message, it determines its actual angle which then determines in which sector it lies. Note that since we only need to calculate $|\sin\theta|$ and not use the angle itself 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. $AG = 45°$.

The algorithm used for localization in this case is similar to Algorithm 2 above. The difference is the proposal that a sensor sends to the leader now contains the sensor's sector information rather than its location. The leader runs the algorithms on all sensor classes (using the expected distance and expected angle for each) to determine the best pair of classes using Eq. 2. From each class a sensor is chosen arbitrarily. Note that with finer granularities, some sectors might be empty and hence their respective classes need not be considered.

Since the target localization uncertainty model that we use depends on both the angle that separates the two sensors 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. Fig. 1(c) shows an example of such a division.

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 $AG = 360°$ in which we will have one class. Also, note that $AG = 180°$ 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.

V. Performance Evaluation

In this section we discuss the result of the experiments used to evaluate 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 effects of changing the maximum number of sensors that can be assigned to a detection task on the detection quality.

A. Simulation Setup

There are two types of deployed sensors: directional acoustic sensors and imaging sensors. We also have tasks of two 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.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 uncertainty of target location of a pair of sensors to a task is found using Eq. 2.

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

$$e_{ij} = \exp\left(\log(P_{FA})\big(1 + \frac{SNR_1}{D_{ij}^2}\big)^{-1}\right) \tag{3}$$

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 [11]. 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. 2). 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 optimal 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.

B. Simulation Results

In Fig. 2, we observe 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 location and fuzzy location schemes. The optimal upper bound is included for comparison. We vary the distance granularity (DG) from 0 to 7 and observe its effect on the fuzzy location performance. The achieved profits increase rapidly as DG increases, but once it reaches 4 there is little further increase. 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

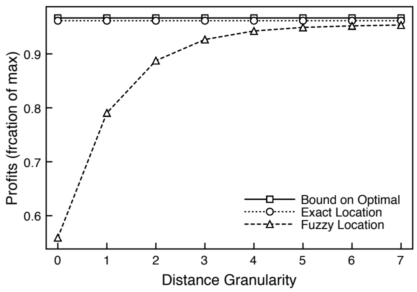


Fig. 2. Detection Performance

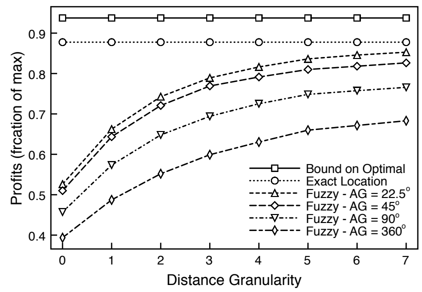


Fig. 3. Localization Performance

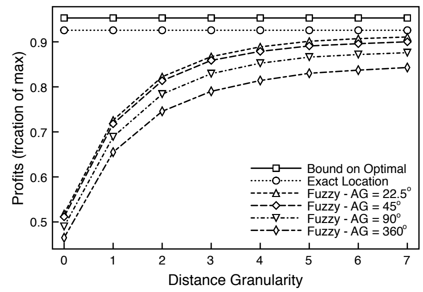


Fig. 4. Overall Performance

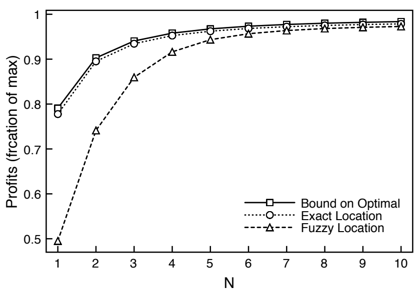


Fig. 5. Effect of Varying N

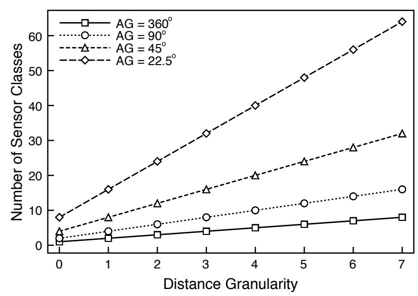


Fig. 6. Computational Overhead

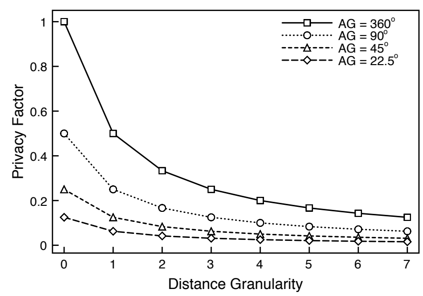


Fig. 7. Privacy

= $360°$, 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 with AG but this increase becomes negligible (less than 1%) when we make AG finer than $22.5°$. We note that the performance of the exact location scheme 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 which sensors are selected compared to detection as it is affected by both the distance and the angle. If the optimal sensors for a task are already assigned to another localization task they will not be available and hence the task will select less than optimal ones. Combining the results of both schemes (Fig. 4), we find that the total network profits are affected by both previous results.

In Fig. 5, the performance of the detection algorithms is measured (in a similar setup) as N increases from 1 to 10. Note that a higher value of N means that more sensors can be assigned to each detection task, which will increase the cumulative detection probability. As expected, the profits of the detection tasks increase as N increases. 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 our cumulative detection function.

C. 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. As expected, when we increase the granularity the number of classes increases as well. The tradeoff between performance and efficiency depends on the number of the nodes that are within the sensing range of the task's location. In our experiments there are on average 31 sensors in that range. For a localization task, if we were to use $DG = 3$ and a $AG = 22.5°$, 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 lead to savings in computational cost. Also, in many cases the generated classes will have no sensors in them (due to their small size) which will make the number of classes to be considered be smaller than the number of possible classes. Note that for tasks that only depends on distance, such as detection, the savings in computational overhead is significant.

Fig. 7 shows a privacy metric for the different fuzzy location granularities. Let N_s be the number of nodes that are within sensing range from the task's location. We use the fraction of N_s which lies in a sector to determine the level of privacy a certain fuzzy granularity can provide. For example, if this fraction is equal to 1 then a proposing sensor could be anyone of the N_s sensors which provides the highest anonymity. If this fraction is $1/N_s$ the task leader can be almost certain of the identity of the proposing sensor since there are no other sensors in that sector. We see that although the privacy level stays relatively high when only distance granularity is increased, it decreases rapidly as we start dividing the circle surrounding the task location into more sectors. Note that the level of privacy is also affected by the density of the network. The more sensors are deployed the higher the value of N_S and hence the better the privacy.

VI. 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 [6], [12], [13]. [12] 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 [13]. 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 [14] and target tracking [15] problems separately. For non-directional sensors, both [16] and [17] 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 [18], 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 [19], 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 [20]. A sensor can be seen as a *resource-constrained robot* as suggested in [21], specifically the problem ST-MR-IA of [20], 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.

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 [22], which addresses the issue of privacy when fusing data coming from sensors that are assigned to multiple event detection tasks, and [23], which describes a data dissemination technique to ensure that the locations of sensors in the network are not learned by an enemy.

VII. Conclusion and Future Work

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] 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*.

[2] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *Journal of Human Computer Studies*, vol. 43(5/6), pp. 907–928, 1994.

[3] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara, "Semantic matching of web services capabilities," in *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web*. London, UK: Springer-Verlag, 2002, pp. 333–347.

[4] A. Kelly, "Precision dilution in triangulation-based mobile robot position estimation," in *Proceedings of Intelligent Autonomous Systems*, Amsterdam, 2003.

[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Colmpleteness*. Freeman, 1979.

[6] A. Bar-Noy, T. Brown, M. P. Johnson, T. La Porta, O. Liu, and H. Rowaihy, "Assigning sensors to missions with demands," in *ALGO-SENSORS 2007*.

[7] S. J. Tutton, "Optimizing the allocation of sensor assets for the unit of action," Naval Postgraduate School, California, Tech. Rep., 2006.

[8] H. Rowaihy, M. P. Johnson, D. Pizzocaro, A. Bar-Noy, L. Kaplan, T. La Porta, and A. Preece, "Exact and Fuzzy Sensor-Task Assignment," Network and Security Research Center, Pennsylvania State University, Tech. Rep. NAS-TR-0106-2009, Jan. 2009.

[9] I. Kadar, "Optimum geometry selection for sensor fusion," in *SPIE 1998*.

[10] L. M. Kaplan and Q. Le, "On exploiting propagation delays for passive target localization using bearings-only measurements," *J. of the Franklin Institute*, vol. 342, no. 2, pp. 193–211, Mar. 2005.

[11] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, 1999.

[12] C. Frank and K. Omer, "Algorithms for generic role assignment in wireless sensor networks," in *SenSys 2005*.

[13] M. P. Johnson, H. Rowaihy, D. Pizzocaro, A. Bar-Noy, S. Chalmers, T. La Porta, and A. Preece, "Frugal sensor assignment," in *DCOSS 2008*, 2008.

[14] J. Ai and A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, Feb. 2006. [Online]. Available: http://dx.doi.org/10.1007/s10878-006-5975-x

[15] Y. Cai, W. Lou, M. Li, and X. Li, "Target-Oriented scheduling in directional sensor networks," in *INFOCOM 2007*, 2007.

[16] N. Ahmed, S. S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in *LCN 2005*, Washington, DC, USA.

[17] M. Hefeeda and H. Ahmadi, "A probabilistic coverage protocol for wireless sensor networks," *ICNP 2007*, pp. 41–50.

[18] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *IPSN '04*, Berkeley, California, USA, 2004.

[19] L. Kaplan, "Local node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 136–146, January 2006.

[20] 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*, vol. 23, no. 9, p. 939, 2004.

[21] 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)*, vol. 36, no. 3, pp. 315–327, 2006.

[22] M. Roughan and J. Arnold, "Multiple target localisation in sensor networks with location privacy," in *ESAS 2007*.

[23] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *ICNP 2007*.