A Survey of Sensor Selection Schemes in Wireless Sensor Networks

Hosam Rowaihy[1], Sharanya Eswaran[1], Matthew Johnson[2], Dinesh Verma[3], Amotz Bar-Noy[2], Theodore Brown[2] and Thomas La Porta[1]

[1]Pennsylvania State University;
[2]City University of New York;
[3]IBM T. J. Watson Research Center

ABSTRACT

One of the main goals of sensor networks is to provide accurate information about a sensing field for an extended period of time. This requires collecting measurements from as many sensors as possible to have a better view of the sensor surroundings. However, due to energy limitations and to prolong the network lifetime, the number of active sensors should be kept to a minimum. To resolve this conflict of interest, sensor selection schemes are used. In this paper, we survey different schemes that are used to select sensors. Based on the purpose of selection, we classify the schemes into (1) coverage schemes, (2) target tracking and localization schemes, (3) single mission assignment schemes and (4) multiple missions assignment schemes. We also look at solutions to relevant problems from other areas and consider their applicability to sensor networks. Finally, we take a look at the open research problems in this field.

1. INTRODUCTION

Sensor networks consist of a large number of small sensor devices that have the capability to take various measurements of their environment. These measurements can include seismic, acoustic, magnetic, IR and video information. Each of these devices is equipped with a small processor and wireless communication antenna and is powered by a battery making it very resource constrained. To be used, sensors are scattered around a sensing field to collect information about their surroundings. For example, sensors can be used in a battlefield to gather information about enemy troops, detect events such as explosions, and track and localize targets. Upon deployment in a field, they form an ad hoc network and communicate with each other and with data processing centers.

Sensor networks are usually intended to last for long periods of time, such as months or even years. However, due to the limited energy available on board, if a sensor remains active continuously, its energy will be depleted quickly leading to its death. To prolong the network lifetime, sensors alternate between being active and sleeping. There are several sensor selection algorithms to achieve this while still achieving the goal of deployment.

The decision as to which sensor should be activated takes into account a variety of factors depending on the algorithm such as residual energy, required coverage, or the type of information required. Sensors are selected to do one or multiple missions. These missions can can be general and related to the function of the network, such as monitoring the whole field by ensuring complete coverage, or more specific and application-oriented, such as tracking a target's movement. At a given time, the system might be required to do multiple missions such as monitoring an event and, at the same time, track a single or multiple moving objects.

In this paper, we survey the recently proposed schemes used for sensor selection. These schemes can be categorized in many different ways. For example, they can be centralized (all processing is done by a single node) or distributed (processing cost is divided upon many nodes). They can also be classified according to the criteria or parameters that decide the selection process, e.g., physical attributes-based, task utility-based, information gain-based, etc. In this paper, however, we categorize the schemes based on the purpose of selection into the following classes:

Send correspondence to Hosam Rowaihy. E-mail: rowaihy@cse.psu.edu

1. *Coverage schemes*: include selection schemes that are used to ensure sensing coverage of the location or targets of interest.

2. *Target tracking and localization schemes*: include schemes that select sensors for target tracking and localization purposes.

3. *Single mission assignment schemes*: include schemes that select sensors for a single specific mission.

4. *Multiple mission assignment schemes*: include schemes that select sensors so that multiple specific missions are collectively accomplished.

The rest of this paper is organized as follows. In Section 2 we define the sensor selection problem. In Sections 3-6 we discuss the different classes of schemes. We present some theoretical approaches as they apply to sensor selection and assignment of sensors to missions in Section 7. In Section 8, we define open research problems in this area. Finally, Section 9 concludes the paper.

2. THE SENSOR SELECTION PROBLEM

The sensor selection problem can be defined as follows: Given a set of sensors $\mathcal{S} = \{S_1, ..., S_n\}$, we need to determine the "*best subset*" S' of k sensors to satisfy the requirements of one or multiple missions. The "best subset" is one which achieves the required accuracy of information with respect to a task while meeting the energy constraints of the sensors.

So, we have two conflicting goals: (1) to collect information of high accuracy and (2) to lower the cost of operation. This trade-off is usually modeled using the notions of *utility* and *cost*:

- **Utility:** accuracy of the gathered information and its usefulness to a mission.
- **Cost:** this consist mainly of energy expended activating and operating the sensors which is directly proportional to number of selected sensors k. Another cost factor that can be considered here is the risk of detecting a sensor which may increase for active sensors especially if wireless communication is used.

The goal of a sensor selection scheme is to select a subset S' of k sensors such that the total utility is maximized while the overall cost is less than a certain budget. For many utility models, this problem is at least as hard[1] as the Knapsack problem which is (weakly) NP-complete.[2] This means there is no polynomial-time algorithm, although there is a pseudo-polynomial algorithm (poly in number of sensors and sensor costs). This is clearly not desirable, especially if we consider a network with a large range of possible sensor costs. Hence, realistic restrictions of the problem have received attention. For example, Isler and Bajcsy[1] assume that utilities have geometric structure and that total cost is either zero or infinity, based on whether $|S'| \leq k$ holds.

3. COVERAGE SCHEMES

In this section, we discuss schemes in which sensors are selected in order to ensure complete coverage of the field. By complete coverage we mean that every point in the filed must be in the sensing range of at least one sensor. We look at schemes that assume static sensors then we look at selection schemes for mobile sensors.

3.1. Selection for Static Sensors

If the sensor nodes are densely deployed, such that there there is redundancy in coverage, then only a subset of sensors need to be active in order to achieve full coverage while the rest can enter sleep mode. This conserves energy and hence prolongs the network lifetime. Selection schemes are used to decide which sensors are to be turned on and for how long. If, during the course of operation, a node fails resulting in a coverage hole, the selection protocols can be rerun to activate one or more of the redundant nodes to restore the coverage.

For example, Perillo and Heinzelman[3] divide the sensor nodes into sets, such that each set is capable of providing complete coverage of the field and only one set is active at a time. The objective is to obtain an

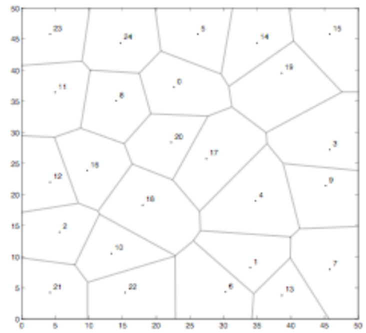

Figure 1. An example of a Voronoi diagram.[5]

optimal scheduling of these sets (i.e., to optimally select which set is active when and for how long) such that the lifetime of the application is maximized and a minimum level of quality of service constraints (such as bandwidth and coverage) are met. This problem is formulated as a generalized maximum flow graph and an optimal solution is found through linear programming (LP).

In their paper, Cardei and Du,[4] divide the sensors in the field into disjoint sets, such that every set completely covers all the targets and only one set is active at a time. Unlike Perillo and Heinzelman's approach[3] here the sets are scheduled in a round-robin order and the focus is on the problem of finding the maximum number of disjoint sets. As the number of sets increase, the efficiency becomes greater because the time interval between two activations for any sensor is longer. They prove that finding this maximum number of disjoint sets, which they define as the *Disjoint Set Cover* (DSC) problem, is NP-complete and they provide a heuristic-based approximate solution. The DSC problem is transformed into a max-flow problem, which is formulated as a *Mixed Integer Programming* (MIP) instance. The output of the MIP is used to compute disjoint set covers in polynomial time.

The main drawback in the above approaches is that it is difficult to implement them in a distributed manner, which is more desirable in a sensor network environment.

In a recent work by Shih et al.,[6] full coverage with minimal sensors is obtained by identifying the redundant sensors and turning them off. Identification of redundant sensors is done using Voronoi diagrams.[7] Voronoi diagrams are a common and useful way of representing the the physical location of the sensors and the coverage thereby provided. A Voronoi diagram divides a plane that includes a number of sensors into polygons such that each polygon contains exactly one sensor and every point inside the polygon is closer to the contained sensor than to any other. Figure 1 shows an example of a Voronoi Diagram.

The algorithms presented by Shih et al.[6] make use of the properties of Voronoi diagrams to find appropriate sensors to activate. These algorithms account for the residual energy of sensors to balance the power consumption and prolong the network lifetime. Initially, all sensor nodes are inactive. An initiator node finds its Voronoi diagrams assuming different neighboring nodes are activated. It then chooses the configuration (i.e. the active node set) that makes the area of its Voronoi cell less than or equal to its sensing area. Finding such an optimal set of neighbors is NP-complete, and hence the authors propose an approximation algorithm which has a complexity of $O(n^2)$.

Lu et al.[8] take a different approach in which they aim to provide k-coverage, which means that every point in the field is covered by at least k sensors. This improves the fault-tolerance of the network. Initially, all sensors are inactive. Before turning on, each sensor waits for a back-off period determined by the amount of contribution they can provide for the coverage. Thus, the sensors are turned on one by one in a greedy fashion, with the sensor with most "contribution" turning on first, then the next one, and so on. The "contribution" or the "coverage merit" is computed based on the probability of detection of an event by that sensor within its sensing area.

The paper by Yan et al.[9] provides a "self-scheduling" scheme, in which time of operation is the only parameter in the selection process. Here, the nodes dynamically schedule themselves while guaranteeing a certain degree of coverage. The sensors are time-synchronized, and each sensor generates a random reference time which is

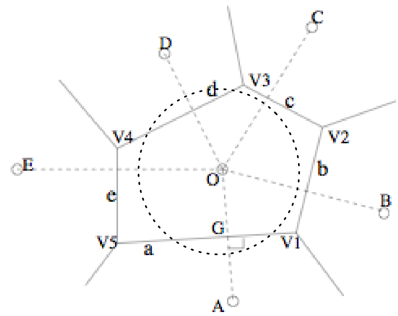

Figure 2. Example of a coverage hole. Node O's Voronoi polygon is greater in size than its coverage area (denoted by dotted circle).[5]

exchanged with its neighbors. Each sensor then establishes its sleep-awake cycle by observing the reference time of its neighbors. Differential coverage is possible by adjusting the node's schedule parameters. If an event requires more coverage, this information is disseminated and the nodes adjust their schedules locally such that there is more overlap, i.e., more sensors stay awake at a time.

3.2. Selection for Mobile Sensors

If sensors are mobile (e.g. placed on robots), more options arise. A sensor node can now move to a new location in order to fill a coverage hole. This makes deployment easier, because even if a random deployment results in incomplete coverage, the nodes can then relocate themselves to ensure coverage. Similarly, if a node fails, then the network can be dynamically reconfigured to make up for it. There are also new challenges posed by mobility for selection schemes because the decision of which node to move has additional energy cost due to movement.

Kwok et al.[10] present a scheme that aims to optimally utilize the limited-range sensor capability and limited-travel capability of a group of mobile sensors (robots) to cover a region completely, after an initial random deployment. The field is modeled as a grid and the problem is formulated as an assignment problem (assigning sensor to grid point). Two variations of this problem exist: (i) to minimize the maximum distance traveled by a sensor and (ii) to minimize the total distance traveled subject to energy (fuel) constraints. The first is an instance of the maximum cardinality (unweighted) matching problem on a bipartite graph which connects the robots to grid points. The second is solved by computing a minimum-weight matching on the graph. We discuss the details of solving the sensor assignment problem using techniques such as matching in Section 7.

Other recent works discuss selection schemes, where the problem is to decide which sensor to move in order to compensate for incomplete coverage due to imperfect deployment or failure of nodes. For example, Sekhar et al.[11] aim to select the best node to substitute for a dead node. Four different schemes, based on different heuristics, are evaluated. The first scheme selects the node with the maximum residual energy. This avoids repeated failures at the same point. The second scheme tries to minimize the migration distance of a node. For each neighbor of the dead node, the maximum distance that it can move is calculated. This distance can vary from 0 to completely reaching the dead node's position itself. The node which has to move the minimum of these maximum distances is selected. This ensures that the maximum area can be recovered, while the migration distance is minimum. The third scheme combines the first two heuristics and selects the node with the minimum value of the ratio (maximum distance to be moved) / (residual energy). While these three schemes move nodes the maximum possible distance towards the dead node (greedy approach), in the fourth scheme, the minimum distance each node has to move is calculated and the node with the minimum of these min-distances is selected, just so that the uncovered area is likely to be covered.

In a recent paper by Wang et al.,[5] a bidding protocol is used for deciding which sensors should move to cover the holes. A mixed network of static and mobile sensors is deployed with mobile sensors being redundant initially. Static sensors then discover coverage holes in the field. Coverage holes are discovered using Voronoi diagrams (Figure 2 illustrates how this can be done). Each mobile sensor has a base price for serving one hole in the sensing field. The price is related to the size of any new hole generated by its movement. The static sensors in the network estimate the size of the coverage holes, and place bids for the mobile sensors accordingly. The mobile sensors choose the highest bids and move to heal the largest coverage holes.

4. TARGET TRACKING AND LOCALIZATION SCHEMES

In this section we discuss the methods in which sensors are selected for the purposes of target tracking and target localization. These schemes can be further classified based on the approach used in the selection algorithm. Here, we consider three categories, (i) Entropy-based solutions, where the selection schemes aim to minimize the entropy of measurement, (ii) Dynamic information-driven solution, where the aim is to maximize the information gain based on the dynamic information gathered and (iii) Mean Squared Error-based solutions, where the aim is to minimize the mean squared error of measurements.

4.1. Entropy-based Solutions

Entropy refers to a measure of uncertainty. The lesser the entropy of some measurement, the more we can be certain of its accuracy. There are several solutions that use this concept. In recent works by Ertin et al.[12] and Liu et al.[13] the authors use the mutual information about the future state and the current node measurement to determine the information gain of the different sensors. A greedy approach is used to solve the sensor selection problem for target localization and tracking. At each round, an unused sensor with an observation that is expected to yield the maximum entropy reduction of the target location distribution is selected. This newly added observation is then used to determine the target location distribution using recursive Bayesian estimation techniques.[14] Sensor selection continues in this manner until the level of entropy of the target location distribution is less than or equal to a predefined value. This value reflects the desired confidence that an application needs regarding the target's location. The goal here is to reach the required entropy level without using more sensors than necessary.

The major problem of this scheme is that it requires a way to effectively evaluate the expected entropy reduction for the different candidate sensors. This is difficult to determine without actually retrieving measurement data from the different sensors. Also, this scheme is centralized in the sense that sensor selection decisions are made by a single node. This is not scalable and incurs a high communication overhead which makes it impractical in many scenarios.

Wang et al.[15] consider sensor selection for target localization and develop a solution based on entropy. However, instead of trying to find the optimal solution using mutual information, which is computationally intensive, it proposes a heuristic that provides a sub-optimal solution but is more efficient. Given a prior probability distribution of the target location and the locations and sensing models of a set of sensors, the method selects an informative sensor such that the collection of the selected sensor observation with the prior target location distribution results in the greatest reduction in uncertainty. The proposed heuristic adds one sensor at a time to reduce the entropy of the target location distribution. Although this solution is more efficient than the ones by Liu et al[13] and Etrin et al.,[12] it is still centralized which limits its scalability.

4.2. Dynamic Information-driven Solutions

The paper by Zhao et al.[16] proposes a sensor selection scheme that is used for target tracking. They consider the problem of selecting a sensor j, such that j provides the greatest improvement in the estimate of a target location at the lowest cost. This is solved as an optimization problem defined in terms of information gain and cost. The goal is to improve: (1) detection quality, (2) track quality, (3) scalability, (4) survivability and (5) resource usage.

The proposed scheme selects a single sensor (the *leader*), that becomes active. The initial selection is made by considering the predicted location of the target. From that point on, after collecting the required measurements about the target, the leader selects the next node that it believes to be the most informative and passes its measurements to it. That node becomes the new leader. This continues as long as needed to track a target.

When deciding on the next leader, the current leader considers the information utility value of candidate sensors. To be practical, this value must be based only on available information such as a sensor's location, its modality and the current belief state. The authors consider two possible definitions of information utility; one based on entropy and another based on distance measure. Although an entropy based definition is mathematically more precise, it is very difficult to compute in a practical setting. This is because it requires knowing a sensor's measurement before making any decision, which is clearly very difficult. With distance based measure, the leader node measures the utility of other sensors based on how far are they located from the estimated target position. This provides a good approximation of the sensor's utility.

One of the drawbacks of this scheme is that its accuracy depends on the quality of the choice of the first leader. If the first leader is not close to the target location, due to an error in prediction, the overall tracking quality will degrade and the whole process might even fail. Also, this scheme selects a single sensor (leader) at a time, so although it may be energy efficient, it might not provide information that is as good as if more sensors are used.

A similar scheme is proposed by Pahalawatta et al.[17] . Here the authors look at the problem of tracking in video-based sensor networks. The main difference from the previous paper[16] is that this paper considers the initialization cost (measured in time) that sensors encounter when performing motion segmentation for target tracking. Instead of activating a single sensor at each time step, a set of sensors is selected such that their total information utility is maximized subject to an average energy constraint.

At each time step, the sensor network will have a processing node active (similar to the leader used by Zhao[16]). This node is the closest node to the predicted target location. The processing node then determines the state of relevant sensors (i.e. active or not) depending on the total information-gain of the set and the energy constraint. After collecting data about a target and having a better estimate of its location, the processing node selects the next processing node from a set of candidates. The new processing node must be closer to the target than the current one. This scheme places a high overhead on the processing nodes.

4.3. Mean Squared Error-based Solutions

Minimizing mean squared error (MSE) is another approach that is used for target localization. The recent works by Kaplan[18,19] discuss several schemes that achieve this in sensor networks for target localization. The system that is considered in these two papers consists of sensors that can estimate the direction of arrival (DOA) of a target using acoustic properties.

In Ref. 18, Kaplan presents a global node selection scheme in which the location of all sensors is used to determine which set of sensors should be active to locate a target. Initially, two sensors are selected as the active set. These two sensors must not be collinear with the target. Selecting the first active set is done by trying all the combinations of sensors and hence is $O(n^2)$. After selecting the initial active set, sensors (from all inactive ones) are added one at a time using a greedy approach. The goal is to minimize the MSE of the target's position. The paper also discusses methods that can be used to improve on the active set by continuously checking if replacing an active node with an inactive one will result in an improvement in localization quality. Because this solution requires global knowledge of the sensors location, this information must be delivered to all sensors. This can be done either by broadcasting this information to all nodes or by using multi-hop routing. Both of these schemes are inefficient in terms of energy and hence this solution can only be used in small-sized networks.

In his next paper, Kaplan[19] overcomes the limitation of the previous scheme by presenting a local node selection scheme called *autonomous node selection* (ANS). In this scheme, the decision on which node is selected is based only on local information about the usefulness of that node relative to the current active set. The usefulness is measured by considering the MSE. The scheme starts by performing an exhaustive search to determine the initial set of active sensors. This is similar to the scheme used in the previous paper. After selecting the initial active set, ANS is used in each node that is within the communication range of the active set. At each iteration, active nodes from previous iteration select k sensors (where k is the required number of sensors) from a list of candidates. The candidate list begins with the initial set but then grow to include previously inactive nodes. The growth of the candidate set is done by calculating the differential utility of all nodes in the active set then determining a threshold value for an inactive set to join the of candidate nodes.

Both of the schemes discussed above incur a high computational overhead especially when performing the exhaustive search to determine the initial set of active sensors which limit their applicability to small networks. They also require the exchange of a large number of messages to find a solution which again affect their scalability.

5. SINGLE MISSION ASSIGNMENT SCHEMES

In a sensor network which must perform a specific mission repeatedly over time, sensors need to be selected such that the mission is accomplished in the most efficient manner. The objective of such selection schemes is to select the sensor nodes that are most useful for the mission. This notion of "usefulness" is quantified using a "utility" value. Th papers we discuss here differ from those discussed in sections 3 and 4 in that they use sensor selection for an "application-layer" task, while those in the previous two sections use selection to perform the basic functionalities of a sensor network, namely coverage and tracking, more efficiently.

Byers and Nasser[20] developed a model for such applications in which the global objectives are defined based on utility functions and a cost model for energy consumption due to sensing and data delivery. In their algorithm, a set of sensors has a total utility function that depend on the number of individual sensors and their placement. The authors use an objective function that maximizes the utility of a sensor network over its lifetime subject to the energy consumption.

In a recent work by Bian et al.,[21] a generic framework in which the application can specify the utility values of the sensors is presented. The goal here is to select a sequence of sensors sets such that the total utility is maximized, while not exceeding the available energy. Alternatively, the framework can be used to look for the most cost-effective sensor set, maximizing the product of utility and sensor lifetime.

6. MULTIPLE MISSION ASSIGNMENT SCHEMES

The methods discussed in this section deal with a multiple-mission scenario. It must be noted that, as in Section 5, by mission, we refer to a specific (application-level) task and not a network level function, such as coverage or data dissemination. These multiple missions may belong to the one big operations, or may belong to multiple operations that the sensor network is responsible for.

Ai and Abouzeid[22] provide a greedy heuristic to cover the maximum number of targets with the minimum number of active sensors. The sensors here are directional and covering each target can be viewed as a different mission. After the initial random deployment, not all targets are covered. Hence, the objective is to change the initial orientations of the directional sensors in order to cover as many targets as possible, while activating as few sensors as possible. This can be formulated as an *Integer Programming* (IP) problem with the number of sensors (n), number of targets (m) and number of orientations available for a sensor (p) as parameters. The authors prove that this problem is NP-complete. So, instead of solving it optimally they provide a greedy heuristic-based solution that works as follows.

Initially, all the sensors are inactive. In each subsequent iteration, each inactive sensor calculates the number of additional targets that would change from being uncovered to covered for each possible configuration. Then, the inactive sensor that maximizes the number of newly covered targets in a particular orientation is activated in that orientation. Thus, the number of targets covered is maximized greedily. A distributed implementation of this approach is also provided.

In a recent paper by Mullen et al.,[23] the authors model the system as a market and explores the advantages of incorporating e-commerce concepts to sensor management. While in e-commerce, the customer wants to essentially drive the process, the conventional sensor management follows a much less capitalistic approach. It produces information "goods" based on pre-defined system goals and priorities. The two main components in this model are the mission manager and sensor manager, which are implemented using genetic algorithms (GA). The mission manager allocates budgets to the application (consumer), based on the different missions involved in the application and their requirements. Using these budget values, the consumer places bids to the sensor manager. Based on these bids, the sensor manager allocates sensors to the missions. This is a centralized system and it is very difficult to implement it in a distributed way because of the complexity of the model and the GA computation involved.

Ostwald et al.[24] use multi-modal sensors and assume that multiple missions may arrive simultaneously. The possible sensor configurations, i.e. which sensor operates in which mode, and the mission utility value for each mission are translated into a bid. The winner is determined using a modified combinatorial auction algorithm.

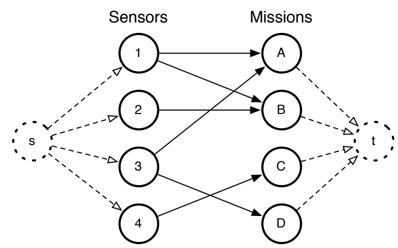

Figure 3. Example of a bipartite graph. Dotted portion shows conversion to flow.

Using this a sensor is assigned to the mission for which it is responsible, and the mode in which it has to operate is selected.

We can also consider the bidding scheme we discussed earlier (by Wang et al.[5]) as a multiple-mission assignment scheme. In this case covering each hole is a single mission. In the same manner, the scheme that is proposed by Kwok et al.[10] can be looked at as a multiple-mission assignment. However, the missions in this case are the grid points to which the robots are assigned.

7. THEORETICAL APPROACHES TO SENSOR-MISSION ASSIGNMENT

We observe that when there are multiple missions that a sensor network is required to do, sensors must not just be selected but assigned to particular missions. A subproblem of the general Sensor Selection Problem is the *Sensor-Mission Assignment Problem*. This problem may be informally defined as assigning sensors to missions in the "best" way, which may depend on the cost of using individual sensors and the requirements of individual missions.

Somewhat more formally, we are given a set of sensors $\mathcal{S} = \{S_1, ..., S_m\}$ and a set of missions $\mathcal{M} = \{M_1, ..., M_n\}$. We may draw this as a bipartite graph, where an edge (S_i, M_j) corresponds to a pairing of sensor S_i with mission M_j. This will in general not be a complete graph, since a sensor may be applicable to only a subset of missions. Each edge (S_i, M_j) may be associated with a utility or cost value. See Figure 3 for an example of this setup. Our goal is to assign sensors to missions optimally (maximizing utility and/or minimizing cost), subject to the constraints we impose. In this section, we define several particular problems using this setup. We survey several classical approaches, as well as recently studied ones, and adapt them to the sensor-mission assignment problem. These approaches are different versions of network flow, semi-matching and online matching.

7.1. Flow-based Approaches

Maximum flow is used to model many kinds of optimization problems that involve the movement of resources from a source a destination. We now consider the bipartite graph representation of the sensor-mission assignment problem presented above and reduce it to a flow problem. By finding the max-flow of such a graph, we can determine the best possible sensor assignment.

7.1.1. Max-Flow

Max-flow models the problem of determining how much of a resource can pass from a source to a destination, given the capacities of each transport segment along the way. More precisely, we are given a directed graph $G = (V, E)$, where V is a set of vertices and E is a set of edges. The vertex set contains two special nodes, the source node s and the sink node t. Each edge is associated with a non-negative integral capacity which is the maximum amount of resource that may pass through it. A flow can be decomposed into a collection of paths. Because edges may in general have different capacities, the amount of flow along any path is limited by the minimum capacity of the edges along that path. Moreover, it may not be possible to use all edges' capacities even if all edges have the same capacity, since a vertex may have more out-going edges than in-coming edges. Given these constraints, we seek a feasible assignment of flow to the edges of the graph that maximizes flow from s to t.

The classical Ford-Fulkerson algorithm[25] solves max-flow in polynomial time. The algorithm starts with an empty flow and then proceeds as follows: as long as there exists in the graph a path from s to t that still has available capacity (an *augmenting path*), increase the usage of the entire path as high as possible. When there are no remaining augmenting paths, stop. Crucially, the max-flow returned by Ford-Fulkerson is integral, i.e., the flow through every edge is an integer.

In the standard max-flow problem, each edge's usage has an upper bound (its capacity) but no lower bound. For some applications, lower bounds can be incorporated. We cannot simply apply Ford-Fulkerson to an instance with lower bounds, since the empty flow may not be feasible. There is a classical solution[26] to this problem as well. Given a graph with upper and lower bounds on the edges, we can transform it into a conventional max-flow graph. This graph will only have upper bounds, but they will differ from the upper bounds of the original. The empty flow is feasible for the new graph, so we can find its max-flow, which is then converted back to a feasible flow for the original graph.

Max-flow models can by useful in solving the sensor-mission assignment problem. Consider the following formulation. We are given the sets of sensors and missions. Each sensor will in general be applicable to some, but not all, missions. Each sensor is associated with a total utility U_i it can contribute. Associated with each mission M_j is a demand D_j, indicating the total utility the mission requires. A sensor may be assigned to multiple missions, distributing its utility fractionally, as long as its upper bound U_i is observed; a mission may have multiple sensors assigned to it, as long as its lower bound M_j is met. What we seek in the feasibility problem is an assignment of each sensor to 0, 1, or more missions, while respecting both sets of constraints.

This problem can be solved efficiently by reduction to max-flow in at least two ways. In both methods, we create a flow-graph by adding two nodes, s and t, as the source and sink (see the dotted part of Figure 3); the capacity of each edge (s, S_i) is set to U_i. For any pairing (S_i, M_j), the capacity of the corresponding edge is infinity if this pairing is allowed and 0 if not. The way the flow graphs differ is in edges directed towards t. One option is to set the lower bound of (M_j, t) to D_j, and then solve max-flow with lower bounds. A second option is to set the capacity of (M_j, t) to D_j, and then solve conventional max-flow. If a feasible solution exists, then max-flow will necessarily find it, since it will be impossible to send superfluous flow to any mission.

7.1.2. Min-Cost Max-Flow

For a given max-flow problem instance, there need not be a unique maximum flow solution. If edges are associated with both capacities and per-unit costs, the *Min-Cost Max-Flow* problem is to find a maximum flow of minimum total cost. There are classical algorithms to find integral solutions to this problem in polynomial time[27] .

The *transportation problem* (a.k.a. the Hitchcock Problem) is a problem that models the min-cost transportation of goods[28] . We are given a weighted bipartite graph of sources and terminals. Edge weight w_{ij} is the unit cost for transportation of goods from S_i to M_j. Each source has a supply P_i and each terminal has a demand D_j. We seek a min-cost way to satisfy all demands using the available supplies. This problem easily reduces to Min-Cost Max-Flow: add s and t to the graph such that the capacity of each edge (s, S_i) is set to P_i, and the capacity of each edge (M_i, t) is set to its (integral) demand D_j.

We now describe a more sophisticated model for sensor-mission assignment. Here, we require unique sensor-mission assignments (i.e. a sensor can only be assigned to a single mission), but relax the nature of the mission demands to become simply the number of sensors that the mission needs. Each possible sensor-mission pair has a cost value. The feasibility problem is to assign sensors uniquely to missions so that each mission receives the number of sensors it demands. The optimization problem is to find a feasible assignment that minimizes the total cost. This formulation is simply the Transportation Problem, with all source supplies equal to 1.

7.2. Matching and Semi-Matching

Now we discuss different ways of representing sensor-mission assignment as a bipartite matching problem. We seek a matching of two node sets representing sensors and missions with the largest cardinality, or the largest weight. In the semi-matching problem, we allow nodes on one side to be used multiple times. This is appropriate here because usually multiple sensors may be assigned to the same mission, but not vice versa.

The *Bipartite Cardinality Matching* problem can be defined as follows: given a bipartite graph, with node sets A and B, we seek a maximum matching in the graph, i.e., a subset of edges of maximum cardinality, under the restriction that no two chosen edges share an end-point. Although there are more sophisticated algorithms[27] , one simple way of solving this problem in polynomial time is by reduction to Max-Flow.

To transform a bipartite matching instance to max-flow, simply add a new node s pointing to each member of A, and a new node t pointed to by each member of B. Set the capacity of every present edge to 1. Now, run a max-flow algorithm which will produce an integral solution. Since all edge capacities are 1, we can read off the matching from the resulting flow.

In the *Weighted Bipartite Matching*, we are given a bipartite graph of two node sets A and B with non-negative weights on its edges. We seek a max-weight matching i.e., a subset of edges of maximum combined weight, again under the restriction that no two chosen edges share an end-point. Finding a *maximum cardinality matching of minimum weight* is essentially the same problem; we can introduce extra nodes so that A and B are equal in size, and we can set the weights of all missing edges to 0. Since weights are non-negative, every max-weight matching will be a perfect matching. By replacing edge weight w_{ij} with edge cost $c_{ij} = W - w_{ij}$, where W is the highest value of all edge weights, we get a minimization problem.

Bipartite *Semi-Matching* is similar to ordinary bipartite matching, except the matching constraint is relaxed on one side: we now seek a subset of edges such that no two chosen edges share an endpoint in A. The problem can be formulated for both cardinality and weighted settings, and can be solved through similar reductions to max-flow, as above.

Both the weighted and unweighted matching models can be applied to sensor networks. Given a bipartite graph of sensors and missions, the presence of an edge (S_i, M_j) indicates that S_i is capable of serving M_j. In a feasible solution, each sensor is assigned to at most one mission. We seek a feasible solution of maximum cardinality. This formulation reduces immediately to cardinality semi-matching, but it is a very coarse model of our problem.

A weighted semi-matching model can also be used to solve the sensor assignment problem. In this case, given a weighted bipartite graph of sensors and missions, each edge (S_i, M_j) is associated with a corresponding a utility value which can be a measure of how useful a sensor is to a mission. The motivation here is that each mission wants to be covered, and different sensors may be capable of covering it, but some might get better quality of information for a mission than others. Each sensor can be assigned to at most one mission. We seek a feasible solution of maximum weight. This formulation reduces immediately to weighted semi-matching, but again it is a very coarse model of our problem. Both the weighted and unweighted versions matching problems were studied by Kwok et al.[10] in the context of sensor-mission assignment.

7.3. Online Matching and Semi-Matching

In the two previous sections, we took for granted that the entire set of missions is given at once, and solve this problem in an offline fashion. In real-life deployments, however, missions will arrive at different points of time. In this section, we study a sequential online model. Each time a mission arrives, we must assign sensors to it, without knowledge of future missions. These decisions are irrevocable, meaning that once a sensor is assigned, it cannot be reassigned to any future mission. In alternative models, we can allow for sensor reassignments or impose a cost on them, but these are not considered here.

Kalyanasundaram and Pruhs[29] study the online unweighted version of the *b-Matching Problem*, and present an optimal deterministic algorithm. In b-Matching, we choose a subset of the edges present in a bipartite graph containing servers and requests. The presence of an edge indicates that a server is capable of serving a request. Each request should be assigned to a server; each server can at most serve b requests. In the online version, we receive requests along with all their edges one at a time; each request must immediately be assigned to a server or be declined service.

The authors give a simple algorithm, BALANCE, whose competitive ratio approaches $1 - 1/e$ for large b. For each request r, BALANCE selects a server, among r's neighbors, which is the least used so far. The usage levels of the servers are kept in balance, as is clear from the name.

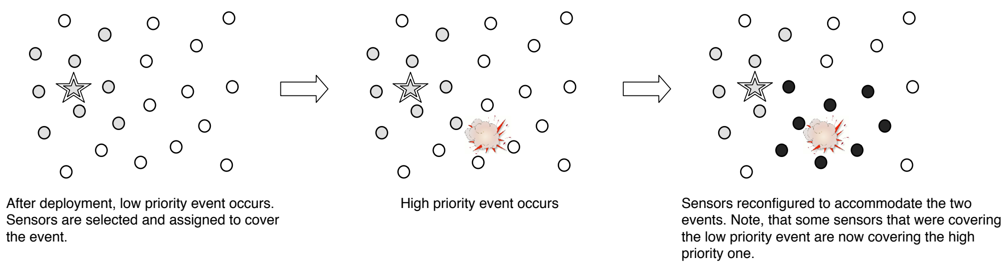

Figure 4. Multi-mission support.

One way to adapt this problem to the context of sensors and missions is to assume that each sensor can serve at most b missions at once, corresponding perhaps to b different sensor modalities. Given this restriction, we then seek a maximal matching in the setting where missions arrive online.

Kalyanasundaram and Pruhs[30] study weighted versions of the online matching problem, which they call Online Min-Matching (minimizing edge weight) and Online Max-Matching (maximizing edge weight). Because both of these are hard in the general case, they restrict themselves to metric spaces which satisfy the triangle equality. An edge weight is then construed as the metric distance between a server's site and a request. Here, k servers need to be matched with r requests such that the total distance traveled is minimized. Min-Matching turns out to be quite hard. The authors give an optimal competitive algorithm, called PERMUTATION, whose competitive ratio is $\frac{1}{2k-1}$, for an instance with $2k$ nodes. For the Max-Matching problem, they give an optimal greedy algorithm with competitive ratio of 1/3.

Since the triangle inequality will be violated by sensor-mission bipartite graphs which include some zero-utility edges, it's not clear how adaptable Max-Matching is to our setting. Although sensor networks exist in the physical (and metric) world, and the utility of a sensor to a target is partly based on distance, there presumably are other factors that influence this as well, such as the modality of information sought.

8. OPEN RESEARCH PROBLEMS

Although the problem of sensor selection has received a sizable attention lately, there are important open research problems that need to be addressed.

To the best of our knowledge, the issue of handling multiple missions that might have different priorities has not been discussed before. A mission can be divided into a set of small tasks. The tasks of multiple missions may not be disjoint, i.e. one or more tasks can be shared between missions and hence there is no need to have duplicate nodes doing the same task. Also, missions may change during the lifetime of the network but some tasks may not, e.g. monitoring an area of interest. Because each node may perform multiple tasks the issue of task-scheduling also arises. Moreover, missions may have different priorities, so handling the assignments of nodes to missions needs to take this into account.

Another issue here is node reassignment. A node that is assigned to one mission may later be reassigned to another mission because it is more useful there. That node then needs to find a replacement. Effects of such node reassignment on missions and its associated cost need to be studied. Figure 4 illustrates an example of handling two missions with different priorities and shows a case in which nodes are reassigned.

The high cost of sensor deployments make multi-modal sensors more interesting. These sensors are able to collect different types of data which means that their contribution to missions will depend on the activated modality. Reconfiguration of the sensors in such a situation must take into account both the task utility and the information gain, along with other factors to choose which modality should be activated and thus poses interesting challenges.

The issue of sensor utility which determines how useful a sensor is to a task is studied in different papers (see Sections 5 and 6). However, there are no realistic models for utility that take the quality of information into account. Currently, most schemes determine the utility of a sensor by its physical attributes such as location.

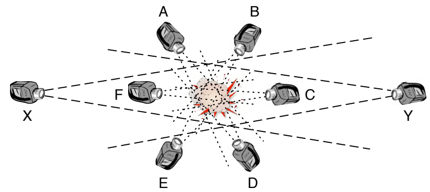

Figure 5. Difference between optimal vs. actual sensor assignment. The optimal set of cameras that should cover the event are the six closest ones (A, B, ... F). However, due to bandwidth constraints, two further cameras (X, Y) are chosen to cover a larger area of the field.

We need to develop quality of information models that take not only physical attributes, but also some sort of semantics into account. For example, a recent work by Bisdikian[31] looked at sensor sampling models and their effects on the quality of information. If we considered video sensors, a video sensor that is the closest to an event might not be the best candidate for selection because its view of the event might be obscured by smoke. Also, although it is embedded in some of the information-gain schemes, the issue of conditional utilities need to be studied in more detail. By conditional utilities we mean how would the selection of one sensor affect the utility of another sensor. For example, in Figure 5, video sensor A may have a high value if selected alone. However, if F is also selected then A's value may be lower since the two cameras have a high overlap in coverage (the area between the two dotted line represent a camera's coverage).

Many of the papers that propose these sensor selection schemes look at the problem from a theoretical stand point. But another issue that needs to be studied is the dynamics of these different schemes and how would they perform in realistic settings in which messages can be dropped and nodes can fail. Also, the different performance aspects of these schemes such as convergence times, communication overhead and how they affect sensor network lifetime need to be studied.

In most schemes, the only cost that is considered when selecting sensors is energy. Although energy might be the most valuable resource in sensor nodes, other resource constraints should be considered. Bandwidth, which is also limited, can be a deciding factor when selecting sensors. For example, an optimal solution to a sensor selection problem might suggest the use of 6 sensors, however, due to bandwidth constraints only 2 can be activated. In this case, the 2 sensors that are selected might not even be from the optimal set of 6 sensors. This example is illustrated in Figure 5.

Finally, an interesting topic is to study whether and how the purpose of selection (e.g. monitoring, answering a query, disseminating data, fault tolerance, etc.) may affect the selection scheme. For instance, the selection schemes for choosing sensors for monitoring purposes may face different requirements and challenges when compared to choosing sensors for answering a query.

9. CONCLUSION

In this paper, we examined the problem of sensor selection in wireless sensor networks. We discussed four different classes of schemes, namely (1) coverage schemes, (2) target tracking and localization schemes (3) single mission assignment schemes and (4) multiple mission assignment schemes. We also looked at solutions to similar selection and matching problems in other fields and discussed their applicability to sensor networks. We believe that there are some important open research problems in this area and we have discussed some of them here.

ACKNOWLEDGMENTS

This research was sponsored by US Army Research laboratory and the UK 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 authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

1. V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005.
2. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Colmpleteness*, Freeman, 1979.
3. M. Perillo and W. Heinzelman, "Optimal sensor management under energy and reliability constraints," in *Proceedings of the IEEE Conference on Wireless Communications and Networking*, March 2003.
4. M. Cardei and D. Du, "Improving wireless sensor network lifetime through power-aware organization," *ACM Wireless Networks* , May 2005.
5. G. Wang, G. Cao, and T. L. Porta, "A bidding protocol for deploying mobile sensors," in *Proceedings of the IEEE Conference on Network Protocols (ICNP '03)*, November 2003.
6. K. Shih, Y. Chen, C. Chiang, and B. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in *Proceedings of the IEEE Symposium on Computers and Communications*, June 2006.
7. F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Computing Surveys* , 1991.
8. J. Lu, L. Bao, and T. Suda, "Coverage-aware sensor engagement in dense sensor networks," in *Proceedings of the International Confernece on Embedded and Ubiquitous Computing - EUC 2005*, Dec. 2005.
9. T. Yan, T. He, and J. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of 1st International Conference on Embedded networked sensor systems*, 2003.
10. K. Kwok, B. Driessen, C. Phillips, and C. Tovey, "Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms," *Journal of Intelligent and Robotic Systems,* , Sept. 2002.
11. A. Sekhar, B. Manoj, C. Siva, and R. Murthy, "Dynamic coverage maintenance algorithms for sensor networks with limited mobility," in *Proceedings of the IEEE Conference on Pervasive Computing and Communications*, March 2005.
12. E. Ertin, J. Fisher, and L. Potte, "Maximum mutual information principle for dynamic sensor query problems," in *Proccedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, 2003.
13. J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *Journal on Applied Signal Processing* , 2002.
14. J. Bernardo and A. Smith, *Bayesian theory*, Wiley, New York, 1996.
15. H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proceedings of the third international symposium on Information processing in sensor networks*, 2004.
16. F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine* **19**, pp. 61–72, March 2002.
17. P. Pahalawatta, T. Pappas, and A. Katsaggelos, "Optimal sensor selection for video-based target tracking in a wireless sensor network," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, 2004.
18. L. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems* **42**, pp. 113–135, January 2006.
19. L. Kaplan, "Local node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems* **42**, pp. 136–146, January 2006.
20. J. Byers and G. Nasser, "Utility-based decision-making in wireless sensor networks," in *Proceedings of the IEEE Workshop on Mobile and Ad Hoc Networking and Computing*, 2000.
21. F. Bian, D. Kempe, and R. Govindan, "Utility-based sensor selection," in *Proceedings of the IEEE Conference on Information Processing in Sensor Network (IPSN 2006)*, April 2006.
22. J. Ai and A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization* , Feb. 2006.
23. T. Mullen, V. Avasarala, and D. L. Hall, "Customer-driven sensor management," *IEEE Intelligent Systems* **21**, pp. 41–49, Mar/April 2006.
24. J. Ostwald, V. Lesser, and S. Abdallah, "Combinatorial auctions for resource allocation in a distributed sensor network," in *Proceedings of the IEEE Real-Time Systems Symposium*, December 2005.
25. L. Ford and D. Fulkerson, *Flows in networks*, Princeton University Press, 1962.
26. S. Even, *Graph Algorithms*, Computer Science Press, 1979.
27. R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows*, Prentice Hall, 1993.
28. C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, 1998.
29. B. Kalyanasundaram and K. Pruhs, "An optimal deterministic algorithm for online b-matching," *Theoretical Computer Science* **233**, pp. 319–325, 2000.
30. B. Kalyanasundaram and K. Pruhs, "Online weighted matching," *Journal of Algorithms* **14**, pp. 478–488, 1993.
31. C. Bisdikian, "On sensor sampling and quality of information: A starting point," in *3rd IEEE Int'l Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2007), a PerCom'07 Workshop*, (White Plains, NY), March 19-23, 2007.