A Market-based Approach to Sensor Management

VISWANATH AVASARALA

TRACY MULLEN

DAVID HALL

Given the explosion in number and types of sensor nodes, the next generation of sensor management systems must focus on identifying and acquiring *valuable* information from this potential flood of sensor data. Thus an emerging problem is deciding what to produce, where, for whom, and when. Identifying and making trade-offs involved in information production is a difficult problem that market-based systems can "solve" by allowing user values, or utilities, to drive the selection process. Essentially this transforms the traditional "data driven" approach (in which multiple sensors and information sources are used, with a focus on how to process the collected data) to a user-centered approach in which one or more users treat the information collection and distribution system as a market and vie to acquire goods and services (e.g., information collection, processing resources and network bandwidth). We describe our market-based approach to sensor management, and compare our prototype system to an information-theoretic system in a multi-sensor, multi-user simulation with promising results. This research is motivated in part, by rapid technology advances in network technology and in sensing. These advances allow near universal instrumentation and sensing with worldwide distribution. However while advances in service-oriented architectures and web-based tools have created "the plumbing" for data distribution and access, improvements in optimization of these distributed resources for effective decision making have lagged behind the collection and distribution advances.

Authors' addresses: V. Avasarala, G. E. Research, Niskayuna, NY 12309, E-mail: (avasaral@research.ge.com); T. Mullen and D. Hall, College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, E-mail: ({tmullen, dhall}@ist.psu.edu).

1. INTRODUCTION

With the advent of small, inexpensive, low-power sensor nodes that can provide sensing, data processing, and wireless communication capabilities, sensor networks can potentially generate huge amounts of diverse data. However, just because the data can be produced, does not mean that it should be. Sensor networks are constrained by limits on sensor "attention" that include limitations on battery power, bandwidth, and the number and type of measurements that the sensor can handle at any one time. In addition, sensor networks can include data collection entities that operate on very different timescales, from human reports to high-speed video-frame collection of images. System users (who may be human, software agents or data fusion processes) each have their own individual tasks and priorities, but share a common sensor resource pool. The sensor manager's job is to efficiently allocate sensors to end-user tasks so as to maximize end-user utility while simultaneously minimizing the cost of collecting, storing, and processing the data. Sensor managers must also consider the interplay between various network resources, weighing tradeoffs between resource constraints such as battery power, bandwidth, and sensor accuracy.

For our current work, we assume that end users belong to a common overarching non-commercial institution. Example application areas for such networks include: (1) network-centric warfare, in which multiple sensing platforms, sensor nets, and individual soldiers with sensors interact to allow rapid tactical situation assessment and threat assessment [11, 12], and (2) monitoring of the environment via ground-based, airborne and space-based sensing systems.

In recent years, information-theoretic approaches have emerged as a promising paradigm for the development of a comprehensive sensor management for multi-task, multi-sensor networks. These techniques rely on optimization of a certain information-theoretic measure like cross-entropy [5, 20] or information gain [6, 27]. Kastella [18] used cross-entropy to determine the optimal search order for detection and classification problem. Kolba et al. [20] extended this framework to permit operation with uncertain sensor probabilities. McIntrye et al. [25, 26] used information gain (the entropy change in environment for a given sensor allocation as the predicate for their hierarchical sensor management architecture. A valuable advantage of information-theoretic approaches is that they are highly flexible and can be easily adapted to new problems. However, information-theoretic sensor management is concerned primarily with scheduling the data-collecting entities (sensors) and other network resources such as energy usage and communication bandwidth have to be considered separately. Additionally, information-theoretic sensor management approaches are myopic in nature, since they optimize some measure of the "quantity of information" obtained during a particular round of schedul-

ing and neglect the "value of information" to the mission objectives.

Non-myopic sensor managers need to solve a highly complex multi-period scheduling problem, since most network tasks like target tracking occur over multiple periods of scheduling. Techniques based on approximate dynamic programming have been developed for this problem. In [5], Castañon considered a multi-grid, single sensor detection problem. Under certain assumptions about the target distributions and probability distribution of sensor measurements, Castañon proved that the optimal allocation policy would be to search either of the two most likely target locations during each round of scheduling. In [6], Castañon considered the problem of dynamic scheduling of multi-mode sensor resources for the classification of multiple unknown objects. To solve this problem, the author proposed a hierarchical algorithm based on a combination of approximate dynamic programming and non-differentiable optimization techniques. Washburn et al. [43] formulate a single-sensor, multi-target scheduling problem as a stochastic scheduling problem and use the Gittin's index rule to develop approximate solutions. Williams et al. [44] consider a single-target, multi-sensor allocation problem with communication constraints and use adaptive Lagrangian relaxation to solve the constrained dynamic programming problem. Schedier et al. [37] have used approximate dynamic programming to allocate gimbaled radars for detecting and tracking targets over a multi-horizon time period. The authors use a three-phase rollout algorithm with the following stages a. generation of candidate sensor allocations b. generation of alternate sensor plans based on results from the first component c. evaluation of the alternate sensor plans to calculate an approximation of the reward function. The details of the implementation of these components are not explained in the original paper. However, for the three-sensor simulation that was presented in their paper, simple heuristics to generate feasible solutions and to evaluate solution performance would have sufficed. These approaches for non-myopic sensor management are pioneering, but substantial further research is required to adapt them to the generic multi-sensor, multi-task sensor management problems.

The network-centric environments that we are interested in also must consider issues related to privacy and communication costs. For example, in network-centric warfare applications, multiple distributed entities accomplish different tasks by connecting decision makers, effectors, and information sources to a common network [27]. Therefore, task information may be localized across individual users. Allowing all required task information to be accessed by an optimization routine is communication-intensive and may violate privacy issues in a distributed environment. Pricing mechanisms can be designed to address privacy issues and minimize communication requirements [42]. Also, market-based approaches offer an inherently distributed mechanism that can compare "apples" and "oranges" using the common numeraire of money, thus reducing communication overhead to the single dimension of price. Under certain assumptions, price systems have been proven to provide the minimum dimensionality of messages necessary to determine Pareto-optimal allocations [16].

For the above reasons, we believe markets based on combinatorial auction mechanisms are a promising paradigm for a comprehensive sensor management. A combinatorial auction is an auction based on exchanging item bundles (e.g., sensor readings+channel transmission) rather than single items. In earlier work on distributed multi-agent sensor management, Lesser et al. [22] surmised that combinatorial auctions could be a promising path for market-based sensor allocation. Shortly after our initial work on combinatorial auctions for sensor management [2], Ostwald et al. [32] also published preliminary work on using combinatorial auctions to find optimal sensor settings in a distributed radar array. The authors optimize a domain-specific and myopic utility function using a combinatorial auction mechanism during each round of scheduling. Resource constraints other than the sensor schedules are not considered.

A generic market-oriented approach to sensor management that is customizable for different sensor network scenarios must address several key issues. The first issue is the mismatch between what users want to buy (e.g., tracking and identifying a target with a specified accuracy) and what network resources are offering (e.g., cpu, battery power, bandwidth, and sensor directivity and operation mode). The problem becomes even more complicated when we consider that different combinations of sensors can be used to track a target, but each combination of sensors may give a different quality of service (QoS). To accurately assess and bid for different sensor combinations, users would need to know the operating parameters of each sensor, and to calculate the QoS for various combinations of sensors. This leads to the second issue of *preference elicitation*, or eliciting user valuations for all possible combinations of resources to different tasks/users. Clearly in this setting, preference elicitation can be computationally and/or communication intensive. For example, if there are n sensors and *m* tasks, $((2^n - 1)m + 1)$ utility valuations must be acquired by the sensor manager from the user to calculate an optimal allocation. The third issue is that *winner determination*, or determining an optimal allocation given all bids, is an NP-hard problem [35]. Although fast algorithms have been developed, thanks in part to ecommerce-driven advances, these algorithms may not always meet real-time requirements.

To address the above issues, we proposed a framework for sensor management using a market-based architecture called MASM (Market-Architecture for Sensor Management) [15, 30, 33]. The sensor manager handles the mismatch between what providers (i.e., sensors) offer and consumers (i.e., end users) want by

Fig. 1. Single-platform market architecture for sensor management.

providing a mapping between user tasks and network resources. Users bid on high-level tasks, while service mapping components convert the high-level user tasks to low-level sensor tasks and finally to actual bids. Once the necessary bids are created, an auction winner determination algorithm computes the final resource allocation. Both the bid formulation and winner determination steps are computationally expensive. Traditionally, humans have been mainly responsible for the bid formulation step, with computational auctions focusing on the winner determination step. One of our contributions has been to develop an approximate algorithm, called Seeded Genetic Algorithm (SGA) [29], that combines these two steps and achieves polynomial run times with a modest loss of optimality. Our earlier work [2, 3, 28, 29] described a high-level framework for MASM, but did not provide any implementation details. This paper describes the implementation of MASM, including the auction protocol, pricing algorithms for network resources and heuristics for avoiding myopic scheduling behavior.

We currently focus on a single-platform design, although we plan to extend this model to multiple platforms and sensor network environments. While we draw from recent advances in ecommerce-based market research, we describe the significant challenges in adapting this approach to reflect typical sensor management environments. We test our prototype sensor management system using a multi-sensor, multi-user simulation framework that models bandwidth and battery power constraints. Comparisons to a priority-based information-theoretic system show that market-based algorithms hold promise for developing comprehensive sensor management systems.

It should be noted that the present approach has limited applicability to smart dust environments, where the number of sensors could be on the order of few hundred thousands. In these environments, the communication costs of relaying sensor measurements to the sink are the dominant costs of network operation. For these environments, a centralized auctioneer cannot be used because of the communication costs involved. Instead, task utility information and price information should percolate to the node level, where individual nodes decide on what actions to perform. Mainland et al. [24] have proposed a price-based decision system for smart dust environments, and Padhy et al. [33] have proposed a utility-based model.

Our paper is organized as follows. In Section 2, we describe the MASM architecture and provide an illustrative scenario in Section 3. Section 4 talks about our continuous combinatorial auction (CCA) protocol developed to minimize communication involved in market operations. Section 5 describes the pricing mechanisms that have been developed to enforce resource constraints in the market. Section 6 introduces an agent learning scheme for market agents to assist users in formulating optimal bidding parameters for different tasks. Section 7 describes our simulation environment, while Section 8 describes our results. We summarize our findings and discuss future work in the last section.

2. MASM

Our current single platform design for MASM is shown in Fig. 1, and derives from the sensor management architecture proposed by Denton et al. [8]. The mission manager (MM) assesses mission-level decisions (e.g., assigning task priority to a mission goal), allocates tasks and budgets to end-users. Within the mis-

sion manager, approaches such as goal lattices (which relate high-level mission goals to lower-level actionable tasks) can be used to measure the criticality of various low-level goals to the overall mission goals, and thus help to determine their respective budgets. Kenneth Hintz and Gregory McIntyre [14] used goal lattices to compute the relative weights of actionable tasks (such as tracking) on the basis of high-level mission goals. Rajani Muraleedharan and her colleagues [30] used goal lattices to determine weights for combining various objectives to optimize routing in a sensor network. Newer developments include dynamic goal lattices [15] that can support more dynamic goal generation from a set of predefined goals.

The sensor manager (SM) acts as a competitive market for buyers and sellers of sensor resources. Sensors and transmission channels are modeled as sellers. Sensors sell their sensor schedule (i.e., their "attention") and transmission channels sell raw bandwidth. End users, or consumers, of the sensor network are interested in higher-end products such as target tracks, environmental searches, and target identification. MASM maps between these high-level tasks and actual resources available in the market using its combined service chart/bid formulator functionality.

MASM provides this functionality in two different modes, either exact service mappings (E-MASM) or approximate service mappings (A-MASM). When the number of sensors is small and the real-time constraints are relaxed, E-MASM mode provides an exact service mapping. In other words, given a task and a set of possible resource combinations that can be used for that task, E-MASM will explicitly calculate the utility of assigning each combination to the task using domain information and task-specific utility functions provided by a service chart. Given n sensors in the network, and m tasks, then in the worst case, $(2^n - 1)m$ bids on resource combinations might have to be formulated. A standard combinatorial auction winner determination algorithm [1] then determines the optimal allocation. One approach used to speed up the bid formulation auctions and the winner determination optimization is to generically restrict the type of bids considered for resource allocation. For example, one could place a bound on the maximum number of items in a bid. Polynomial algorithms for bids with certain special structures [35] are available. However, imposing generic constraints on bid types can lead to market inefficiency. An alternate approach is to use domain-specific knowledge to intelligently restrict the number of resource bids formulated. For example, if the types or locations of sensor resources that can be used to accomplish a particular task are limited, the combinations of resources that need to be considered can be reduced.

When the number of sensors is large and real-time constraints are strict, explicit mappings are no longer feasible, and the A-MASM mode, with approximate service mappings, is used. Instead of the bid formulator

explicitly formulating combinatorial bids for each user task, MASM searches the search space of useful sensor combinations directly using a polynomial, anytime evolutionary algorithm [29].

3. SCENARIO EXAMPLE

To illustrate the MASM system, we describe a simple scenario with two users and two sensors. Assume that a particular user is interested in searching and identifying reconnaissance drones approaching from a certain region R_1. Let the task that the user wants to accomplish using the sensor network resources be the reduction of entropy of the probability distribution of the existence of reconnaissance drones in region R_1 to less than a threshold ε_1 (task A).

The user should submit a bid to MASM in the following format:

(type: *search/identify*
entity: *reconnaissance drone x*
region: R_1
quality: (*entropy* $< \varepsilon_1$)
price: P_A)

where P_A is the user's bid price.

Assume that another user is interested in estimating accurately the position of an already identified slower moving reconnaissance drone. Let the task that this user is trying to accomplish using the sensor network resources be the reduction of the track uncertainty (as measured by some reasonable metric such as state vector covariance error) to less than ε_2. The user should submit a bid to MASM in the following format:

(type: *track*
entity: *reconnaissance drone y*
quality: (*covariance error* $< \varepsilon_2$)
price: P_B)

where P_B is the user's bid price.

Assume that two sensors, a forward looking infrared (FLIR), or infrared camera, and a radar are available to the SM for accomplishing these tasks. Note that any two sensors will generally have different abilities to locate and identify targets depending on characteristics such as environment conditions, target characteristics, and target-sensor geometry. In other words, certain sensors, or combinations of sensors, can provide more or less value for task completion, and thus the bid values for different sets of sensors may also vary. To express this, MASM generates combinatorial bids in exclusive-or format for each user's task as shown in Table I during each of scheduling. The exclusive-or format ensures that task A can win either bid 1 or bid 2, but not both. For each bid in Table I, the bid amount is represented using the format $P_{U,S,t}$ where U is the task identification, S is the given sensor combination number and t indicates the scheduling round. This notation is used to indicate that bid prices depend on sensor combination and user

Task A's XOR bids		Task B's XOR bids	
Bid 1	Bid 2	Bid 3	Bid 4
(type: *search/id*	(type: *search/id*	(type: *track*	(type: *track*
entity: *reconnaissance drone x*	entity: *reconnaissance drone x*	entity: *reconnaissance drone 3*	entity: *reconnaissance drone 3*
region: R_1	region: R_1		
sensors requested: *FLIR*	sensors requested: *FLIR and Radar*	sensors requested: *FLIR*	sensors requested: *FLIR and Radar*
quality: (*entropy* $< \varepsilon_1$)	quality: (*entropy* $< \varepsilon_1$)	quality: (*covariance error* $< \varepsilon_2$)	quality: (*covariance error* $< \varepsilon_2$)
price: $P_{A,S_1,1}$)	price: $P_{A,S_2,1}$)	price: $P_{B,S_1,1}$)	price: $P_{B,S_2,1}$)

task. The value of $P_{U,S,t}$ is calculated by the SM using the bid prices of the original consumer bids (see Section 4 for details). Until the tasks are complete, the SM monitors the progress of the tasks and adjusts the bids accordingly.

We describe the methodology used by MASM to generate the bids for resources during each round of scheduling in the next section. The combinatorial auction winner determination algorithm is then used to calculate the optimal resource allocation, given the MASM bids.

4. CCA PROTOCOL

In this section, we describe our continuous combinatorial auction (CCA) protocol. The CCA protocol was designed to increase the computational and communication efficiency of our market-based scheduling algorithm. Since MASM uses discrete time slots to schedule resources, most user tasks, like tracking a target, require acquiring resources over multiple time slots. Each round of scheduling can either occur periodically at fixed times, or randomly. A simplistic allocation of resources across multiple time slots can occur in two ways: i) Users send in a bid that covers resource needs across multiple time slots. The SM updates the schedule upon receiving each new user bid. ii) Users send in a bid for the current time slot only. After each round, users update their requirements based on what was received in the last scheduling round, and send in an updated bid for the next time slot.

The first approach is computationally expensive. Determining the optimal scheduling for n sensors over a time horizon T is exponentially complex in n and T. Clearly, the second approach is communication intensive. We designed the CCA protocol to avoid the communication and computation requirements of using markets for sensor management. Below, we describe the CCA protocol in detail.

CCA executes each of the following steps (except initialization, which is executed once at the start of operations) during each round of scheduling.

4.1. Initialization

Auctioneer initializes the prices for all the resources. It informs the users about the set of tasks that it will accept bids for.

4.2. Update Bids

At the beginning of each round, users can i) send new bids, ii) remove their current bids from the auction, iii) modify the parameters of their existing bids. User bids are of type $\langle t, p \rangle$ where t is the task description, which includes the task type, and final task quality desired by the user and p is the price that the user is willing to pay. For example, the task description for a bid to track a target x such that the trace of the covariance matrix of the target estimate is less than 0.001 is as follows:

(type: *track*
entity: *target x*
quality: (*trace of covariance matrix* < 0.001))

The auctioneer predefines the set of tasks that the user can bid for and the bid format. Here we make the standard assumption that a scalar valued "quality" measure can be calculated using the various task parameters. For example, for target tracking, the trace or the determinant of the covariance matrix can be used as one measure of target track quality [13, 31, 36]. However, under certain circumstances, it is advantageous to use more elaborate task descriptions (see [17] for related discussion). For example, the requirement to identify a target with a given level of specificity and level of confidence may require extensive models of multi-sensor performance in complex observation environments. These are application-specific, and would need to be developed for the particular application being considered.

4.3. Update User Requests

Auctioneer accepts new bids or updates to existing bids during each round of scheduling. If the auctioneer receives no message regarding a particular bid, the bid

stays active and competes for resources in the current auction round.

4.4. Resource Bid Formulation

Since user bids are for high-level tasks, the auctioneer needs to compose bids for actual resources from them. This responsibility is handled by the bid-formulator module in E-MASM (explicit formulation of bids for resources is not required in A-MASM). Let the user bid on a high level task T at time t_i with price P_T. A high level task, such as tracking a target to a required accuracy, might require resources over multiple rounds of scheduling. For each time slot t, the auctioneer constructs bids on each resource set, S, that can be allotted to task T. The auctioneer needs to calculate the price associated with resource set S for task T during each round of scheduling, based on the user bid price P_T. To correctly align task priorities with bidding price, we have devised a novel mechanism for resource price determination. For a resource set S, the auctioneer computes the bid price for a resource set as the percentage of the user task completed by the resource set given the current task status. To determine the percentage of task completed by a resource set S, we cast the problem in terms of optimally scheduling sufficient readings from a canonical sensor A to meet the quality of service (QoS) task parameters. Let the task T require on average n_a consecutive schedules of the standard sensor A to be completed (task is considered complete, when the task quality meets the QoS threshold in the task bid). Suppose a resource bundle S is used when the task quality is q, and the expected number of standard sensor readings required is reduced to $\bar{n}_a$ Then $f_{S,T,q}$ or the percentage of the task completed by resource set S when the current task quality is q, is equal to the percentage savings in the required number of canonical sensor readings.

$$f_{S,T,q} = (n_a - \bar{n}_a)/n_a.$$

There is a possibility that $f_{S,T,q}$ is negative ($n_a < \bar{n}_a$). For example, in spite of allocating sensing resources, the inaccuracy associated with a target estimate might increase with time. To avoid negative prices for bundles of resources, the bid price for $P_{S,T,q}$ (bid price for allocating resource set S to task T during a particular round of scheduling when the current task quality is q) is calculated as

$$P_{S,T,q} = P_T * f_{S,T,q} - P_T * f_{\phi,T,q}$$

where ϕ is the null set.

The auctioneer uses this price to prioritize between different tasks during a particular schedule. However, this price is not charged to the user. Users are charged only at the end of a successful task completion, or if they choose to withdraw a bid before the task could be completed by the SM (see the round termination step). Calculation of n_a and $\bar{n}_a$ can be made faster, by storing
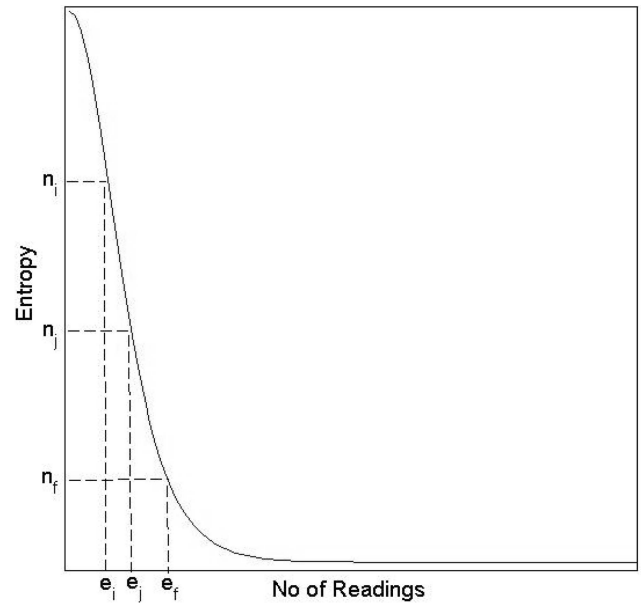

Fig. 2. Illustration of calculation of bid prices for resource bundles using QoS chart.

task specific performance data for the canonical sensor as QoS charts. For example, consider a user bid for searching a particular grid for potential threats, when QoS is measured in terms of entropy. A sample QoS chart is given in Fig. 2, and shows the expected fall in entropy with standard sensor readings. Let a user bid specify a task for reduction of entropy of a particular grid from e_i to e_f. If a resource bundle S is expected to reduce the entropy from e_i to e_j after the next reading, then

$$f_{S,T,e_i} = (n_j - n_i)/(n_f - n_i).$$

Creating a QoS chart for a sample task is illustrated in Section 7.

Resource formulation is the slowest link in CCA protocol, but heuristics can speed this step up. For example, for certain tasks, it may be feasible to use only a few kinds of resources. Thus, if a task requires only acoustic data, then only the acoustic radar sensors need to be considered. However, in the worst-case scenario, the number of bids is exponential in the number of sensors. This is clearly infeasible in case of large systems and thus E-MASM is not scalable to large systems, limiting its effectiveness. The A-MASM formulation avoids explicit bid formulation, and hence maintains polynomial run-times, both in number of resources and users by using our SGA algorithm, an approximate polynomial-time algorithm. For a detailed description of this algorithm, and a comparison of A-MASM and E-MASM time performance, see [34].

4.5. Resource Allocation

Resource bids, obtained from step 3 are exclusive-OR bids in the form $\langle S_1, P_1 \rangle xor \langle S_2, P_2 \rangle \ldots xor \langle S_n, P_n \rangle$. This bid indicates that during the current round of

scheduling, the user is willing to pay a price P_1 for the resource bundle S_1 and a price P_2 for S_2, but only the maximum of P_1 and P_2 for $S_1 \cup S_2$. The auctioneer needs to translate these bids to OR bids, so that standard integer programming formulations [1] can be used. An OR bid of the form $\langle S_1, P_1 \rangle or \langle S_2, P_2 \rangle \ldots or \langle S_n, P_n \rangle$ indicates that the user is willing to pay $P_1 + P_2$ for the bundle $S_1 \cup S_2$. This can be done by the addition of phantom items [9]. The idea is to translate an exclusive-OR bid B-xor of the form $\langle S_1, P_1 \rangle xor \langle S_2, P_2 \rangle \ldots xor \langle S_n, P_n \rangle$ into a B-OR bid of the form $\langle S_{1b} \cup b, P_1 \rangle or \langle S_2 \cup b, P_2 \rangle \ldots or \langle S_n \cup b, P_n \rangle$, where b is a phantom item. The phantom item b ensures that a maximum of only one bid from the OR bids can be labeled as winner (since each item can be allocated to maximum of one bids). Once the bids are translated into the "OR" format, the winner determination problem becomes a standard integer programming (IP) problem, that can be characterized as

$$\max \sum_{j=1}^{n} p_j x_j \; s.t. \sum_{j|i \in S_j} x_j \leq 1 \quad \forall \quad i \in \{1 \ldots m\}$$

where x_j is 1 if the bid is accepted in the final allocation and 0 otherwise. The IP problem can be solved using a commercial software package like CPLEX. The winner determination problem is NP-hard [18] and in theory, this resource allocation step could prove computationally expensive for E-MASM. Performance of the IP formulation depends greatly on the characteristics of the probability distribution from which bids are generated. For example, the time taken by a problem with one thousand bids and hundred items on a 2.8 GHz Pentium IV processor varied between 0.001 seconds to 5000 seconds, depending on the bid distribution. However, we found that the problems generated by the sensor network simulation are relatively easy for CPLEX (see Section 4).

4.6. Round Termination

The auctioneer updates the costs of resources expended on a particular bid by adding the price of its allocated bundle. For each user bid b, the cost of resources allocated to the bid is updated as

$$C_b = C_b + \vartheta_{s_i}^t$$

where S_i is the bundle allocated to b during the tth round of scheduling and $\vartheta_{s_i}^t$ is the price of the bundle S_i during the tth round of scheduling. This is calculated as the sum of the prices of the individual resources comprising S_i (see Section 5 for explanation of resource pricing).

Also, the auctioneer verifies if the task quality required by each user bid was achieved. When a task is complete, the bids for that task are removed from the auction and the corresponding user is sent the completed task details. The bidding user is charged the minimum of his bid price P_b or the cost of resources spent on the task by SM, C_b.

$$payment_b = \min(C_b, P_b)$$

where $payment_b$ is the fee charged to the user, P_b is the bid price, and C_b is the total cost of the resources allocated to the bid. This fee structure ensures that no user is charged more than their bid price for any task. When $C_b < P_b$, the user has a positive surplus of $P_b - C_b$. An alternate fee structure that divides the surplus between the SM and user is as follows:

$$payment_b = \min(C_b, P_b) + H(P_b - C_b).*\gamma*(P_b - C_b)$$

where $H(x)$ is the Heaviside step function and γ is the percentage of surplus given to SM. If the user withdraws a bid before the task demanded in the bid is completed, the SM charges the user

$$payment_b = \min(C_b, f_{b*}P_b) + H(f_{,b,*}P_b - C_b).$$
$$*\gamma*(f_{,b}*P_b - C_b)$$

where $f_{,b,}$ is the percentage of the task that is already completed by the SM. This is calculated using QoS charts (as described in the resource bid formulation step). This fee structure has been designed to mitigate the impact of dishonest user behavior (see Section 8 for details).

Finally, the auctioneer updates the prices of the resources based on the demand in the current round. We describe how prices are updated in the next section. The auctioneer then updates the resources about their schedules during the current round and sleeps until the next round of scheduling begins.

5. PRICING MECHANISMS

To set prices for individual resources, we use a pricing protocol similar to the tatonnement process. *Tatonnement* is an iterative procedure for finding equilibrium prices based on the search parameter (e.g., price or quantity) [21, 34, 40]. The price adjustment process starts with an auctioneer communicating an arbitrary price set to the users. The users compute their demand for the first good at the given prices and communicate it to the auctioneer. Depending on whether the aggregate demand for the first good is positive or negative, the auctioneer either increases or decreases its price. This process continues until a price at which aggregate demand for the first good equals zero is reached. This process is then repeated for the second good and so on. At the end of the first cycle, only the last good is guaranteed to have a zero demand, but assuming gross substitutability (i.e., when the price of good j goes up, there is a positive increase in the demand for every other good by each user) the price set arrived at after each cycle is closer to equilibrium than the previous one. More refined algorithms using partial derivatives of the demand functions have been developed to search for equilibrium in parallel [38, 45]. Though the gross substitutability assumption is often violated (as in sensor networks), the tatonnement process has been found to give satisfactory results [7].

To use the tatonnement process in MASM, we model the supply and demand functions for a resource at a particular price. MASM estimates these functions using the current resource usage rate. Prices for individual resources are initialized to zero during the sensor network initialization. After each round of scheduling, the prices (ϑ_S^{t+1}) for the resource S for the next round of scheduling are calculated based on the current usage rate of the resource (r_S^t) and the available usage rate of a_S^t.

$$\vartheta_S^{t+1} = \max(0, \vartheta_S^t + \tau * (r_S^t - a_S^t))$$

$$\vartheta_S^0 = 0$$

where τ is the constant which determines the rate at which prices are updated.

The definition of r_S^t and a_S^t is dependant on the resource being modeled. For example, for sensors, we have used the available battery power. Let sensor A be endowed with initial battery power b_i and assume that Sensor A needs to be available for a total operating time of T. At time t, if the available battery power is b_t, then

$$r_A^t = (b_i - b_t)/t \quad \text{if} \quad t > 0, \quad 0 \text{ otherwise};$$

$$a_A^t = (b_t)/(T - t) \quad \text{if} \quad t < T, \quad 0 \text{ otherwise.}$$

Ideally, the tatonnement process would update the price of one resource, run the winner determination algorithm to find the new demand for resources, then conduct price updates for the second resource, and so on. However, because of time and communication constraints, all the price updates are conducted simultaneously using the current rate of utilization, during every round. We expect that the results between the two approaches will not be very different, since the usage rates are moving averages and do not vary significantly based on the usage during the current time slot.

6. AGENT LEARNING

In MASM, the SM accepts bids only on a set of pre-defined tasks. The user agent is responsible for decomposing the high level tasks or goals that it has a utility for into a sequence of SM acceptable subtasks on which it can bid. Also, the user agent has to assign appropriate priorities or bid prices to these sub-tasks, so that its overall performance is optimized. Appropriate assignment of priorities to these sub-tasks has a significant impact on agent performance in the market. As an initial exercise, we experimented with agent learning that uses a simple, greedy Widrow-Hoff based learning to optimize bid parameters based on current market data. For reasons of brevity, the details of this approach are not provided here, but readers are directed to [42]. A more rigorous learning method will be the subject of future research.

7. SIMULATION ENVIRONMENT

A simulation environment consisting of a two-dimensional search area involving multiple targets,

multi-user and multiple sensors was developed for testing MASM and comparing its performance with other sensor management approaches. The design of the sensor network, including the communication channel, is inspired by the DARPA sensor network implemented to carry out research in sensor management domain [22]. These sensor networks are more platform-based and differ from the emerging field of "smart dust," where the sensor network could consist of millions of sensors.

Our simulation environment is representative of the types of sensors, communications resources, and mission objectives for a tactical military environment. The various sensor parameters we used are based on realistic sensor models and are obtained from [26]. While this is a basic scenario, with a limited number of sensors and targets, it is representative of the types of non-commensurate sensors that would be available for other applications such as environmental surveillance and crisis management systems (e.g., for homeland security). We make a few simplifying assumptions about sensor models since our main purpose is to test SM performance rather than absolute fidelity to field conditions. Below we describe our simulation model.

7.1. Users

Users consist of a set of software market agents that search for and destroy targets. These agents have the ability to attack any position within a range of r meters and any target that falls within γ meters of an attacked position is destroyed. The agents are not provided with any sensing resources and they depend on the sensor network for obtaining information about the environment. They bid for sensor resources during each round of scheduling and update their status based on information provided by the sensor manager. Initially, agents move along the simulation area with constant velocity v_c, searching for targets. They use the sensor network's resource to search for potential targets and if the probability of target existence within their range exceeds a threshold $p_{\text{threshold}}$, initialize target tracks. Once a target track is initialized, agents can attack a target if the 99% confidence interval of the target's position is less than γ meters. Hence, they are required to track the target to the required accuracy before attacking it. This is again accomplished by buying sensing resources from the sensor network. Agents are assumed to have a utility u_t for destroying a target. To divide the overall utility into utilities for search and track tasks, agents initially use equal priorities. During the simulation run, agents update the search to track budget ratio using the learning method, mentioned in Section 6.

7.2. Targets

Targets are randomly distributed throughout the search area. They move randomly along the city roads with constant velocity v_t, corrupted by a Gaussian white

TABLE II
Sensor Characteristics used in Simulation

Sensor No.	Type	Range	Bearing	Axis	P_{D}	P_{FA}
1	Doppler	90 m $\pm$ 10%	$1° = 6\sigma$	x	0.95	0.001
2	Radar	30 m $\pm$ 10%	$0.1° = 6\sigma$	x	0.95	0.001
3	FLIR	NA	$0.1° = 6\sigma$	y	0.99	0.001
4	ESM	NA	$1° = 6\sigma$	x	0.5	0.01
5	IR	NA	$100~\mu\mathrm{rad} = 6\sigma$	x	0.99	0.01
6	Radar	30 m $\pm$ 10%	$0.1° = 6\sigma$	y	0.95	0.001
7	Doppler	90 m $\pm$ 10%	$1° = 6\sigma$	y	0.95	0.001
8	Radar	30 m $\pm$ 10%	$0.1° = 6\sigma$	y	0.95	0.001

noise with variance Q. Two different types of targets are modeled (T_1 and T_2). Users have greater utility for destroying T_2 targets. Only T_1 targets were used in the simulation experiments, unless otherwise specified.

7.3. Sensors

The simulation models several different kinds of sensors, including sensors that provide range and bearing, bearings-only sensors, and Electronic Support Measure (ESM) sensors. Measurements of two bearings only sensors, which are not located at the same position, can be combined to create both range and bearing estimates and can be used as a pseudo-sensor. A formal way of modeling sensors is to model their properties, such as bandwidth, wavelength, duration of waveform, signal power per pulse, receiver noise strength diameter of radar aperture. A much simpler modeling technique, in which a sensor's characteristics are characterized by three parameters, its probability of detection P_{D}, probability of false alarm P_{FA} and bearing [6], is used in this simulation. The simulation environment has eight different sensors of five different types that are located on two different platforms orthogonal to each other. The operating characters of the various sensors are given in Table II. Both the platforms are 100 km away from the search area. Since the distance of the sensors from the simulation area is large, small angle approximation $s = r*d\theta$, where s is the length of area that falls under the sensor's beamwidth, $d\theta$ is a beamwidth of the sensor in radians and r is the distance of the sensor platform to the city (100 km). For a detailed description of the sensor modeling techniques adopted in the simulation, refer to [25, 26]. Each sensor has a battery with e_{initial} units of energy. For the purpose of brevity, all the sensor tasks are assumed to cost zero energy, except the task of transmitting messages. The energy spent in transmitting a message of m bytes over a distance of d meters is calculated as $\dot{\alpha}d^2m$ where $\dot{\alpha}$ is a constant (see Table III).

7.4. Communication Channel

For communication purposes, a RF communication channel with capacity C is used. All the messages are assumed to be of uniform size M bytes. The communication protocol used is a contention-based protocol (like CSMA/CD) where each agent with a message to communicate senses to see if the channel is busy and transmits if it is not. If two entities start transmitting at the same time, they back off and wait for a random amount of time. The time taken for communication, via this channel, for a fixed number of messages is stochastic. SM enforces the bandwidth constraint by restricting the probability that the time taken for communication is greater than t_{com} to less than $\beta\%$.

7.5. Sensor Manager (SM)

Since the number of sensors is not large, E-MASM formulation is used. Bids on two types of tasks, search and track, are accepted by SM. The QoS for search tasks is in terms of entropy and for the tracking tasks, the norm of the estimate covariance is used. To create the QoS mapping shown in Fig. 2 for the detection task, the following procedure is used. Let the initial probability of target presence in a particular cell be $\pi_0 = 0.5$. (with $\pi_0 = 0.5$, the cell has the highest possible entropy). The initial entropy of the grid H_o is calculated as

$$H_o = g(\pi_0) \qquad \text{where} \quad g(\pi) \text{ is defined as}$$
$$-\pi * \log(\pi) - (1 - \pi) * \log(1 - \pi).$$

Assume that the canonical sensor A with probability of detection λ_{D} and λ_{FA} is used for verifying the presence of the target in this cell. Assume that a target is present in the cell. Then, the estimated probability π_n^t of target presence in the cell after n consecutive readings of A, can be calculated using Bayesian analysis. Similarly, let the estimated probability after n consecutive readings by A, if the target is not present in the cell be π_n^{nt}. The expected entropy of the cell after n consecutive readings of A is

$$\bar{H}_n = \pi_0 * g(\pi_n^t) + (1 - \pi_0) * g(\pi_n^{nt}).$$

The plot of $\bar{H}_n$ vs. n is used as the QoS mapping for the detection task.

7.6. Information-Theoretic Sensor Manager (ITSM)

To compare the performance of MASM, we needed an alternate sensor manager that can handle multiple

TABLE III
Parameter Values used in Simulation

Parameter	Value	Description
Total no of time slots	500	Total number of resource allocation schedules
n_c	5	No of consumers
n_t	10	No of targets
n_{t_2}	2	No of targets with offensive capabilities
v_c	50 mps	Velocity of consumers
$p_{\text{threshold}}$	0.99	Detection threshold
v_t	50 mps	Velocity of targets
Q	0.01	Variance of Gaussian white noise of target motion
r	50 m	Maximum distance that consumers can attack
γ	1.5 m	Radius of destruction around attacked position
ρ	99%	Required confidence interval length of target's position estimate
u_t	1.0 m	Utility for destroying a target
τ	0.005	Tatonement factor
C	2 Mbps	Bandwith of communication channel
M	1 Kb	Size of communication message
t_{com}	2 millisec	Maximum time allowed for communication
β	0.01	Required probability that time taken for communication is greater than tcom
α	1 pJ/bit/m2	Energy required to send messages per unit distance per unit message size
e_i	2.5 KJ	Initial energy of sensor batteries

heterogeneous tasks and multiple heterogeneous sensors. As explained in Section 1, the currently available approximate dynamic programming based approaches were not directly applicable to this problem without substantial additional work. For this purpose, we implemented an information theoretic sensor manager (ITSM). Hint and McIntrye [25] used information gain (the entropy change in the environment for a given sensor allocation) as the predicate for their hierarchical sensor management architecture. The amount of information gained can be measured by the change in entropy prior to and preceding a sensor measurement. ITSM calculates the information gain, associated with each possible allocation and schedules the resources as per the allocation with the highest information gain. To ensure that ITSM considers the "value of information," we optimized a weighted measure of information gain, instead of relying on the raw information gain. We used the formulation in Kalandros et al. [17] for priority based information-theoretic based sensor management. Instead of multiplying the information gain by the corresponding task weight, the authors use the formula $I'_{s,t} = I_{s,t} + \log(\theta_t^t)$ where $I'_{s,t}$ is the weighted information gain, $I_{s,t}$ is the information gain obtained from allocating sensor suite s to task t and θ_t^t is the priority of task t. A key issue in the use of ITSM is the priorities that need to be assigned to the various tasks. We exhaustively tested the performance of ITSM by varying the track and search budget ratios and found that the optimal user performance was obtained when a track to search budget ratio of $0.9 : 1$ was used. To enforce the bandwidth constraint, ITSM does not consider allocations that require bandwidth, which has more than 0.01% chance of crossing the t_{com} limit. The expected time taken for a particular bandwidth consumption was determined by

using monte-carlo simulations. ITSM does not model energy constraints and these are handled in the experimental setup as explained in the results section.

8. RESULTS

8.1. ITSM vs. MASM Experiments

Information-theoretic sensor managers schedule sensors to minimize the entropy of the environment. However, incorporating battery power constraints into ITSM is not straightforward since most systems either use ad-hoc metrics or else do not address power constraints explicitly. Instead of initially testing against multiple ad-hoc solutions, we compare the ITSM system using two sets of experiments: 1) all the energy requirements of the sensor network are assumed to be zero, and 2) ITSM and MASM consume the same amount of energy for different tasks (as shown in Table III), but ITSM does not use any explicit policy for allocating battery power across the mission. Once a sensor has exhausted its battery, it is not considered in future allocations. In the simulation, the user's primary goal is to destroy as many targets as possible. Therefore, we evaluate sensor management performance by calculating the average number of targets destroyed by ITSM and MASM, as shown in Fig. 3. The left bar graph shows experiments where energy constraints are zero, while the right bar graph shows experimental results when energy constraints are enforced. In both cases, MASM was more successful in meeting user objectives (i.e., in destroying the targets) than ITSM. However, in the second set of experiments, some of MASM's success can be attributed to a better energy enforcement policy and it is not clear from these experiments whether MASM will outperform ITSM for
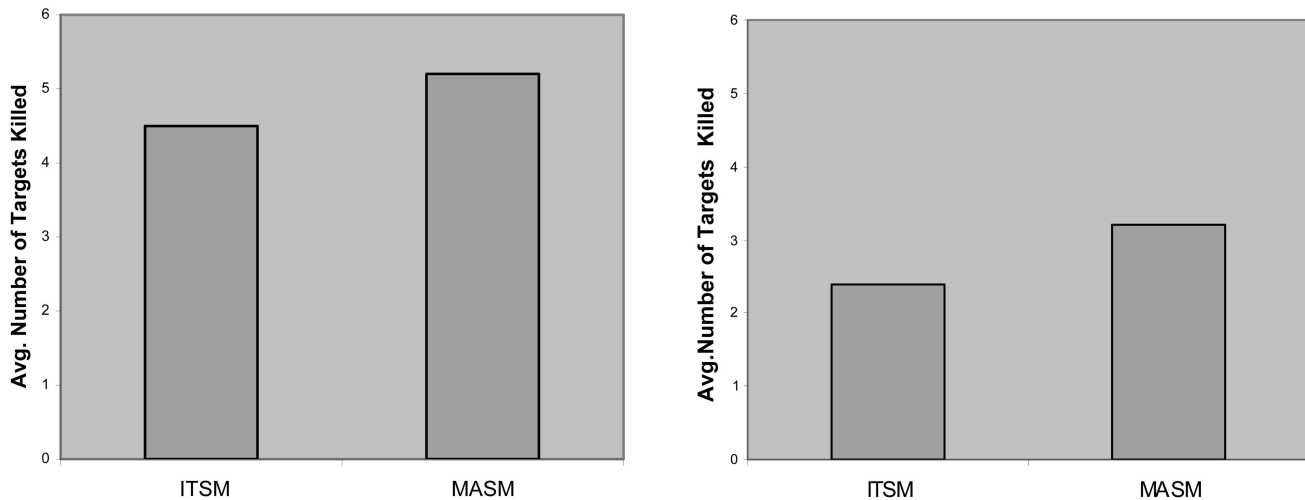
Fig. 3. Comparison of MASM with ITSM (averaged over 100 runs). Left bar graph shows experiments where energy constraints are neglected. Right bar graph shows experiment results when energy constraints are enforced.

any given energy usage policy. We note however, given that MASM does achieve a higher number of targets killed even when the sensor network battery power is "free" for both systems, that this would appear to indicate that MASM's superior performance is not entirely due to a better energy enforcement policy.

Two reasons that MASM outperforms ITSM in meeting user objectives may be that MASM 1) operates to maximize user utility rather than to maximize the information content, and 2) uses prices to prioritize tasks. We discuss these two reasons below.

1) *MASM acts to allocate resources to maximize user utility (as indicated by their bid prices).* Since a user's utility depends on how well the allocated resource set contributes to the user's goals, market-based resource allocation automatically takes goal-related parameters directly into consideration. On the other hand, ITSM concentrates on maximizing information content, neglecting the value of the information to the goals. The priority-based ITSM does a better job than the standard ITSM at incorporating user goals (as priorities) but the system itself has no means of considering a task's progress toward the goal. As an example of why tracking progress toward a goal can be useful, consider the following simple scenario. A single sensor is used to track two targets T_1 and T_2 with equal priority simultaneously. For the first reading, ITSM gets the most information content from tracking T_1, then for the second reading, ITSM gets the most information from tracking T_2. When the confidence interval necessary to attack these two targets is tight, ITSM will never get enough sequential readings to lower the uncertainty sufficiently and will oscillate between the two targets. On the other hand, MASM has equal likelihood for choosing either of the two targets, in any round of scheduling. This happens because the fraction of task completed per reading for either of the target tracks remains constant. Therefore, MASM finishes the tasks in a finite time.

To ensure that our intuition about ITSM vs. MASM was accurate, we implemented the following simple experiment, based on the above scenario, where a single sensor tracks the two targets T_1 and T_2 simultaneously. Target motion is simulated by the equation:

$$x_T(t + 1) = x_T(t) + w_T$$

where $x_T(t)$ is the target position at time t and w_T is white Gaussian noise with constant covariance $Q = 0.005$. Targets can be attacked and destroyed if the 99% confidence interval of their position is less than $\beta_{threshold} = 0.5$ unit. The sensor makes one measurement during each time period, and the measurement equation is:

$$z(t) = x(t) + v(t),$$

where $x(t)$ is the state vector, and $v(t)$ is zero mean white noise with constant variance, $R = 0.03$. Let the initial uncertainties in the position of T_1 and T_2, β_1 and β_2 are equal to 1 unit.

Two sensor-scheduling approaches were implemented. The first approach schedules the sensor to maximize the information gain from the sensor measurement. The second approach schedules the sensor to maximize the utility of measurements, which is defined as the inverse of the total number of sensor measurements required for bringing the targets to threshold uncertainty. The optimal measurement is determined using exhaustive enumeration techniques. The change in uncertainty of the two approaches is shown in Fig. 4 and Fig. 5. ITSM oscillates between the two targets without collecting enough information on any one target long enough to successfully destroy either. The above experiments give an unfair advantage to the utility-based approach since the optimization routine considers multiple sensor schedules simultaneously. In spite of this bias, these experiments offer an insight into the handicap suffered by ITSM due to its inability to take goal related parameters, like $\beta_{threshold}$, and utility-based calculations
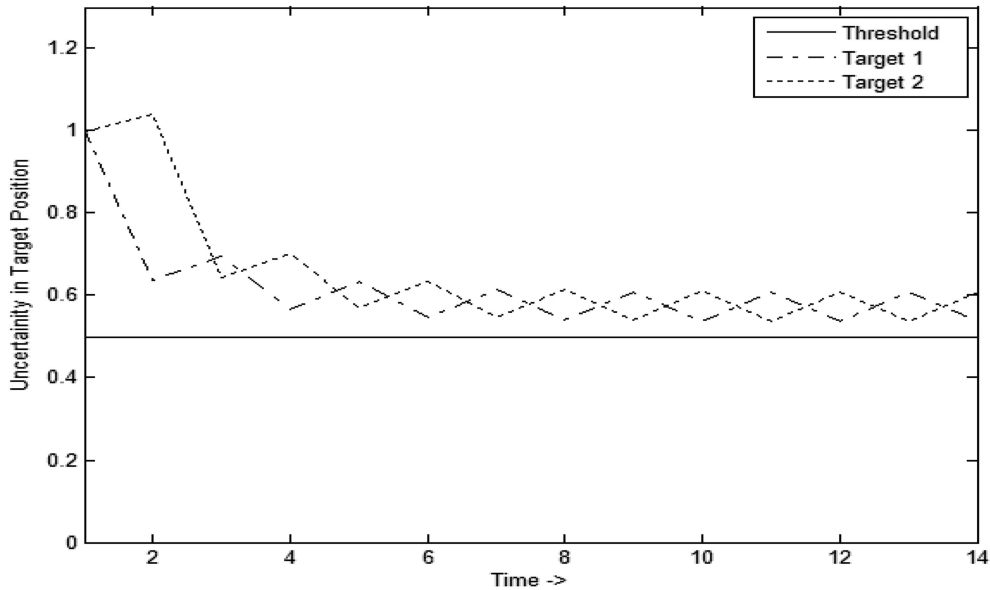
Fig. 4.   Change in uncertainty of target tracks, while using an information-theoretic approach.
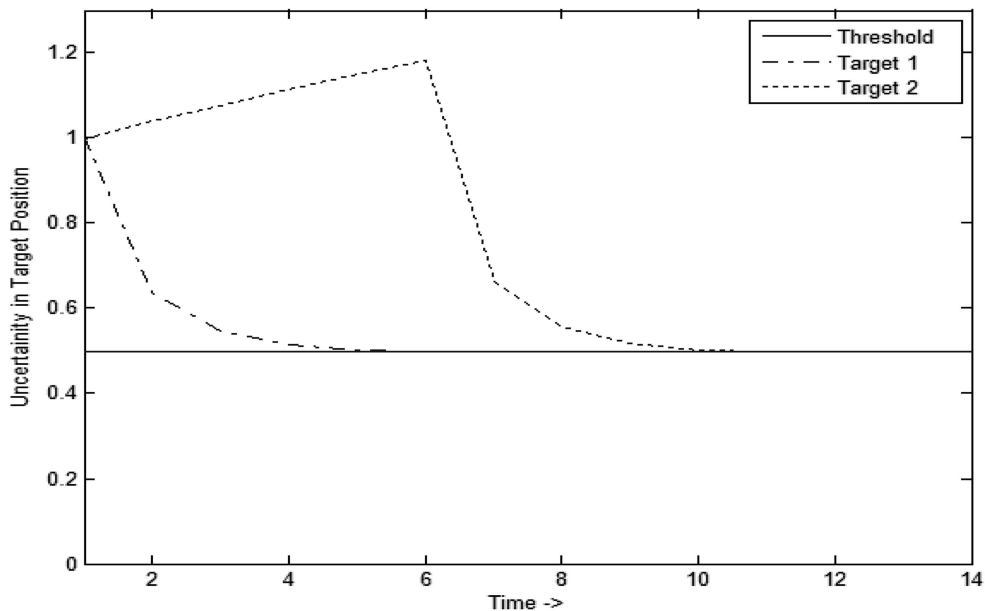

Fig. 5.   Change in uncertainty of target tracks, while using a utility-based approach.

directly into consideration. On the other hand, markets provide a principled way to take the utility of a given allocation to high-level goals directly into consideration during scheduling. These results are analogous to those obtained by Castañon [5]. Operating under some assumptions, Castañon considered the problem of determining the optimal sequence of measurements of a single sensor such that the probability that at-least one target is successfully located in a multi-cell grid is maximized. Results demonstrated that the greedy approximation to the optimal solution performed vastly better than an algorithm based on entropy minimization.

2) *MASM prioritizes using prices.*   Another advantage of MASM may be due to its use of prices to prioritize tasks while ITSM schedules sensors so as to op-

timize the information gain from the environment. Although both ITSM and MASM used the same weights to prioritize between tasks in the environment, price-based task prioritization has some inherent advantages. This is because a price-oriented approach has the ability to implicitly reserve resources for future use by high priority tasks, even if no high priority tasks are currently in progress. For example, consider a situation where the first user is tracking a target and the rest of the users are in search mode. Both MASM and ITSM give highest priority to the track task. The first user has a high-budget for a track-bid and bids accordingly. However, during the tracking task, the prices associated with the sensing resources increases since the rate of their battery power usage during tracking is high (refer
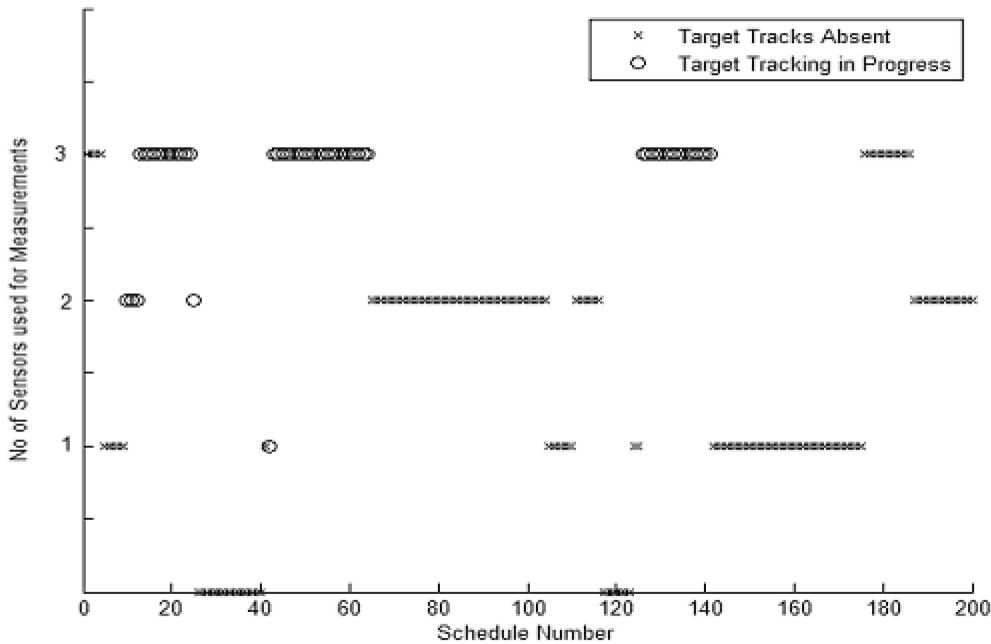
Fig. 6. A comparison of the number of sensors used for measurements, based on whether target tracks are currently in progress or not.

to Section 5 for a description of how prices vary with rate of utilization). After the tracking task is completed and only when detection tasks are in progress, prices of the sensor schedules would have increased. Consequently, sensors will be used at a slower rate during the detection phase, effectively reserving sensors for future higher-priority tasks. However, ITSM has no method of prioritizing between two tasks, except when both the tasks are currently in progress. Fig. 6 shows the number of sensors used during different rounds of scheduling using MASM, where the number of sensors used when tracking tasks are in progress is higher than the number of sensors used when only detection tasks are in progress. When only detection tasks are present, a significant percent of sensors are resting, thereby preserving their battery power for future use.

8.2. MASM-Specific Performance Measures

In addition to our comparison with ITSM, we discuss other MASM-specific performance measures, namely managing resource constraints, task deadlines, scalability, and surplus sharing. The purpose of these experiments is to show how the various "knobs" of a market-based approach can be adjusted to affect performance.

8.2.1. Resource Usage

As explained in Section 5, MASM uses a tatonnement process for enforcing resource constraints such as battery power constraints' using current and available rates of utilization. Fig. 7 shows the price variations of the first three batteries using a tatonnement rate $\tau = 0.005$. Figs. 8 and 9 show the fall in the energy of

the first three sensors' battery with successive schedules with $\tau = 0.005$ and $\tau = 0$ respectively. It is clear that tatonnement process is successful in ensuring uniform usage of sensors and in keeping them functional till the end of network operation. For bandwidth also, a similar procedure is adopted. The supply of capacity is constant and is proportional to t_{com}. During round $t + 1$, if the price of channel is p_t units/sec and bandwidth consumption is w, then the bound of the $1 - \dot{\beta}$ one-sided upper confidence interval of the expected time taken to communicate, η, is calculated based on monte-carlo simulations. The demand at p_t is proportional to η. Values of η for different bandwidth usages are calculated at the start of network operations and stored. The price of a channel during the current round is

$$p_{t+1} = p_t + \tau(\eta - t_{com}).$$

The price updates for process works as a soft constraint on channel capacity. That is, if the channel is too congested, then prices of the channel will increase till demand for channel capacity decreases. However, it is possible that during certain schedules, the actual time taken for communication is more than the prescribed limit. Figs. 10 and 11 show the time taken for communication for two sample simulation runs with $\dot{\tau} = 0.005$ and $\tau = 0$ respectively.

For the run in Fig. 10, the number of time slots when time taken to communicate crossed the specified threshold is 5. This is within the 0.01% tolerance limit specified by the SM. For the run in Fig. 11, the number of time slots where time taken to communicate crossed the specified threshold is 124. After the 250th schedule, all batteries are exhausted, and the time taken to communicate drops to zero.
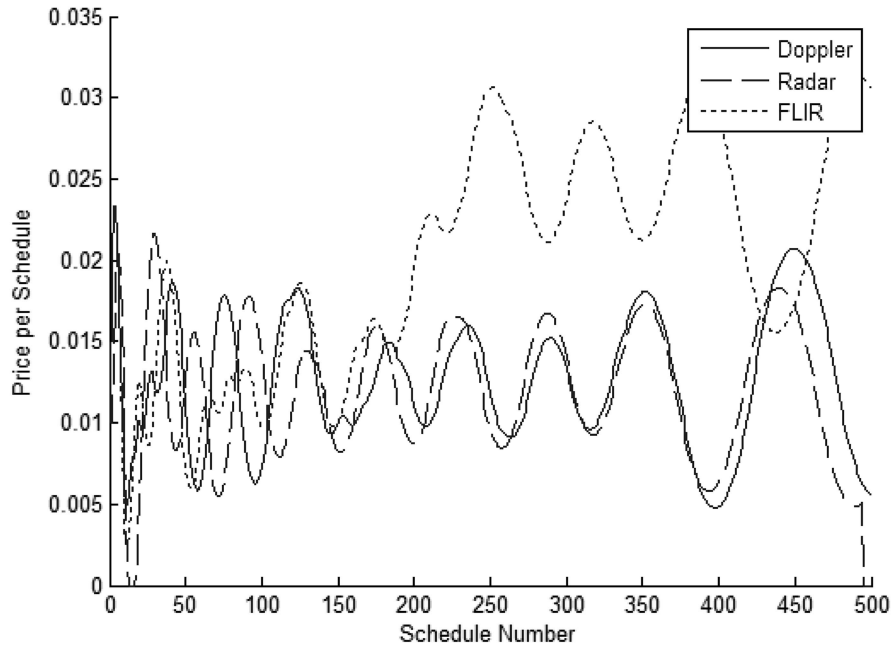
Fig. 7.   Price variation of the first three sensors with schedule number for a sample run with tatonnement $\dot{\tau} = 0.005$.
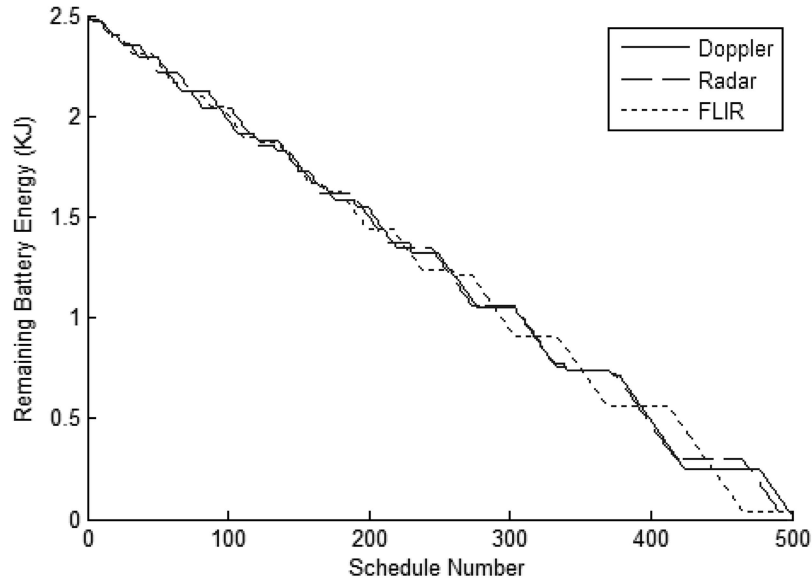

Fig. 8.   Energy usage for the first three sensors for a sample run with tatonement $\dot{\tau} = 0.0059$.

8.2.2.   Task Deadlines

For some high-value targets, users have a strict deadline to destroy targets after initiating target tracks within t_{kill} schedules. We conducted experiments where a certain fraction of the targets are high-valued. We implemented a user policy of increasing track bids by a factor k, if the detected targets are high-value. For our initial experiments, we used a constant value of 3. However, in the future, optimal k values could be calculated by using the market data. Users recorded an average track time of 7.1 schedules for T_2 versus an overall average of 15.3 schedules, showing how markets can be used to enforce task deadlines.

8.2.3.   Scalability Analysis

For the current simulation environment, the IP-based E-MASM formulation can optimally solve problem sizes of up to 10,000 resource bids within a threshold of 10 cpu-seconds on 2.3 GHz PIV processor. Using the SGA-based A-MASM approach, problem sizes of up to 50,000 resource bids can be solved to more than 98% optimality under the same conditions. Larger networks can be accommodated by the approximate technique by compromising on the final optimality. This capability indicates that MASM can be used for fairly large networks, since spatial restrictions often mean that even if a sensor network has thousands of sensors, only a few can be used for a given task at a time.
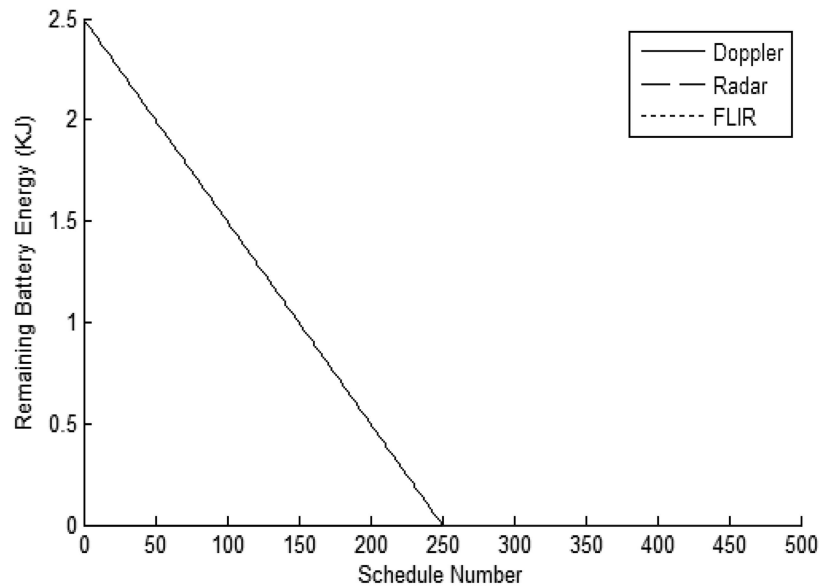
Fig. 9. Energy usage for the first three sensors for a sample run with tatonement $\dot{\tau} = 0$.
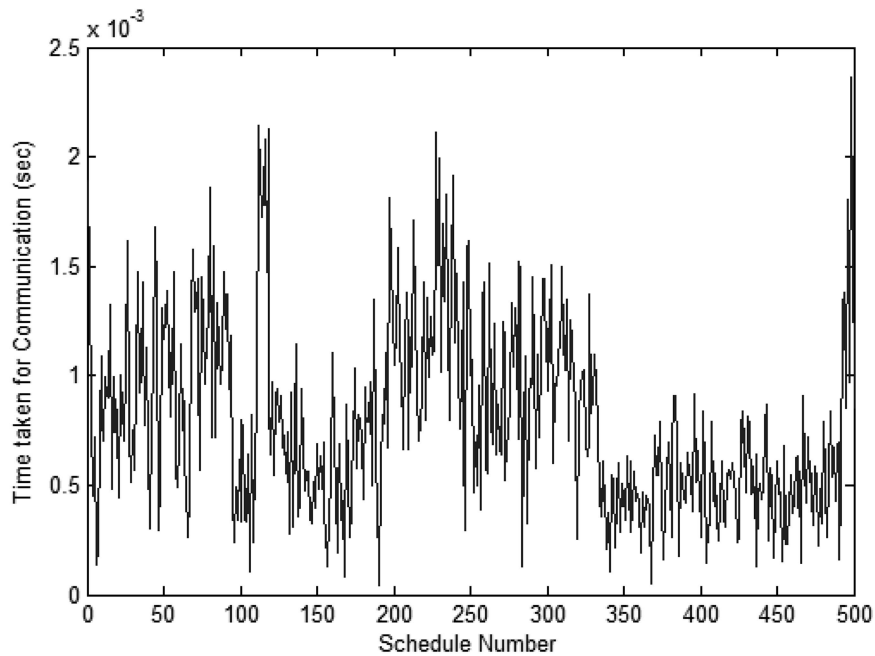

Fig. 10. Time taken for communication for a sample run vs. schedule number for a sample run with tatonnement $\dot{\tau} = 0.005$.

8.2.4. Surplus Sharing

It is possible that sensor networks could have users in a non-cooperative environment, where each agent has an interest only in maximizing its own utility. For example, two different organizations could be sharing the same sensor network resources. Such scenarios require an incentive compatible auction methodology, to make truth revelation the dominant strategy, and thus to make the allocations optimal. For example, a payment mechanism based on General Vickrey Auctions (GVA) [39] might be used to make truth revelation a weakly dominant strategy. GVA involves computation of $n + 1$ winner determination problems for every combinatorial auction to calculate the agent payments. In addition to

the computational complexity, the unique pricing mechanism used by the CCA protocol to ensure real-time response precludes direct adaptation of GVA mechanisms.

To understand the effect of strategic bidding, a preliminary analysis can be conducted by formulating some simple strategic bidding formulations and conducting simulation experiments. For example, Walsh et al. [41] have analyzed the effects of strategic bidding on a combinatorial auction based supply chain formation algorithm. It is important to note that the difficulty of analyzing the effects of strategic bidding actually undermines the benefits of lying about true utilities for users. An added advantage with MASM is that there is disengagement between the users and the actual sensor network.
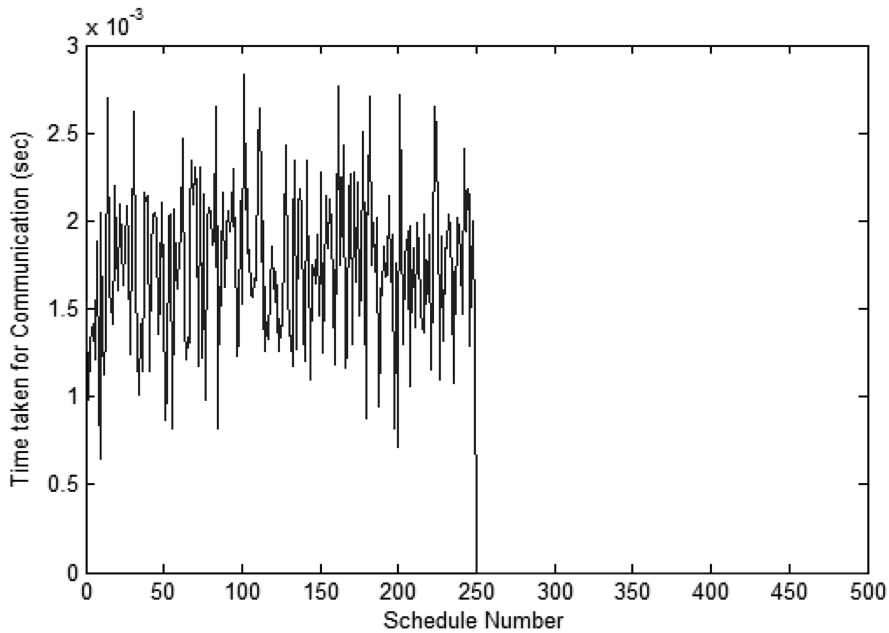
Fig. 11. Time taken for communication for a sample run vs. schedule number for a sample run with tatonnement $\dot{\tau} = 0$.

Though the users use the sensor network, the various network parameters including the prices of individual resources (which might indicate network bottlenecks) and the actual network parameters, like position of sensors, etc. is invisible to the individual user. Hence, the threat presented by malicious entities that have clandestinely gained access to use the sensor network is minimal. To analyze the effect of strategic bidding, it can be assumed that agents play Bayes-Nash strategies [19]. However, calculation of Bayes-Nash equilibria is difficult, except for the simplest of markets. An easier method for analyzing market behavior is to devise a reasonable strategic bidding policy for users and study resulting market behavior.

A simple strategic bidding policy for MASM users is to overstate their task utilities. To understand the logic behind this policy, the pricing policy of CCA should be considered. If a MASM user bids a price P for a particular task, the price it has to pay for resources allocated to its bid is not directly based on P. Instead, for any resource bundle allocated to the task during resource allocation, the user usually pays only the sum of the prices of the resources comprising the resource bundle (see round termination step in CCA). Therefore, a user that overstates its utility has the advantage of getting preferential treatment during resource allocation, while not having to pay any additional value for resources as compared to honest users. To analyze the impact of strategic bidding, experiments were conducted where a certain number of agents overstated their utility by a factor, k. The number of targets successfully destroyed by the users during the simulation experiment reflects the global performance of the market-based resource allocation. An individual user's performance is measured by its surplus defined as the difference between the to-

TABLE IV
Market Performance with Strategic Agent Behavior

Number of strategic agents	0	2
Honest consumer surplus	1.22	−.012
Strategic consumer surplus	NA	2.3

tal utility it obtained from destroying the targets and the total price it paid to SM for buying resources during the simulation.

Two sets of experiments were conducted. In the first set of experiments, all the users bid honestly. In the second set of experiments, two out of the five users overstated their utilities by a factor of two. The average surplus achieved by the honest and strategic agents is shown in Table IV. As expected, strategic agents benefited from overstating their bid prices and their average surplus increased from 1.22 to 2.3. The average surplus of the honest agents has decreased from 1.22 to −0.012 as a result of strategic bidding.

As shown by the preliminary analysis, CCA protocol encourages strategic behavior in a non-cooperative environment. However, the benefits of strategic bidding can be mitigated by using surplus sharing mechanisms where the SM charges the users a fixed percentage of their surplus on each task bid. For example, a variant of CCA, CCA-SS (Combination Auction Algorithm with Surplus Sharing) has been implemented where users are levied an additional charge of fifty percent of their expected surplus as calculated from their task bids. Under this mechanism, the overall surplus to the strategic agent decreased to −1.83. That is, they fare worse than the honest users.

Though surplus sharing provides an effective mechanism that works as a disincentive against overstating

task utilities, detailed experimentation is required to analyze the complete implications of strategic behavior for some particular environment.

9. CONCLUSION

Market-based approaches provide a valuable framework for designing systems that must consider complex tradeoffs in their decision-making. Although much work has been done on individual aspects of market-based systems (e.g., auction algorithms, agent bidding strategies, etc.), there is very little work on developing a complete system in a complex real-world domain such as sensor management. Therefore, one of our contributions is to assess the design implications and components involved in building such a system. To address the issues that in the past have prevented the use of market algorithms use for sensor networks, we have developed techniques including auction mechanisms for aligning scheduling with mission objectives, approximate techniques for handling computational complexity (A-MASM), pricing mechanisms for enforcing resource constraints and surplus sharing mechanisms to reduce the impact of strategic behavior. We have also shown the system's efficiency in a simulation environment, by comparing with a weighted information-theoretic sensor manager. A crucial advantage of the proposed mechanism is its flexibility. The proposed mechanism can be easily adapted to an alternate sensor network scenario without much additional work by suitably creating QoS charts for relevant network tasks (see Fig. 2) and by adapting price equations to reflect utilization of the appropriate network resources (see Section 5). This contrasts with the current approaches based on approximate dynamic programming that are based on domain-specific and cumbersome formulation.

We are currently working on implementing these mechanisms using real-world sensor data. We are also working on developing an alternate non-myopic sensor management approach for comparison with MASM. Two possible choices are i) Maximum marginal return (MMR) [4] sensor management approach that is extended to incorporate non-myopic scheduling behavior and ii) an approximate dynamic programming based approach, similar to [37, 43, 44], that can handle multi-sensor, multi-task sensor management problems. In future, we plan to extend our market-based approach to smart dust environments, which do not have a centralized sensor manager. We also plan to extend our infrastructure to more effectively allocate information goods and develop a market-based situation assessment component. Current auction algorithms are generally designed to handle the allocation of tangible goods. However, we must adapt these e-commerce algorithms to deal with information goods in the sensor-fusion domain. Information goods (e.g., observations/reports),

and the sensors/processes that generate them, may be shared between agents to effectively complete compatible tasks, where applicable. For example, if two users are engaged in the same sub-goal, and want the same information, then a single commodity can be communicated to both agents and will satisfy both of them. We plan to make use of the current research in digital auctions [10], but will need to apply it appropriately to our domain.

We would also like to develop a market-based situation assessment component that learns valuable situation assessment cues from the market bids and price information in the system. The situation assessment that users perform typically relates only to their immediate surroundings and pertains to local information only. A global perspective can be obtained by observing the overall market trends in the sensor manager's situation assessment module. For example, a sudden increase in the volume of bids from the users in one particular region of the environment could suggest an impending enemy attack in that region. Current prices can convey information about when resources are in high demand and/or scarce. As part of this effort, collaborative filtering approaches, similar to those used by Amazon [23], could mine information from a combination of goal and bid behaviors to detect strategic patterns. Eventually, one could imagine a sensor management system that recommends a new information product (e.g., a target track) based on what previous users in similar situations have selected.

REFERENCES

[1] A. Andersson and F. Ygge
     Managing large scale computational markets.
     In *Proceedings of the Thirty-first Hawaiian International Conference on System Sciences, Software Technology Track*, Jan. 1998, 4–13.

[2] V. Avasarala, T. Mullen, D. L. Hall and A. Garga
     MASM: A market architecture for sensor management in distributed sensor networks.
     In *Proceedings of the SPIE Defense and Security Symposium*, Mar. 2005, 5813–30.

[3] V. Avasarala, H. Polavarapu and T. Mullen
     SGA: An approximate combinatorial auction winner determination algorithm for environments with strict real-time constraints.
     In *Proceedings of the IEEE-WIC/Association for Computing Machinery (ACM) International Conference on Intelligent Agent Technology* (IAT-06), 571–578.

[4] C. F. Camerer
     Behavioral Game Theory.
     Princeton, NJ: Princeton University Press, 2003.

[5] D. A. Castañon
     Optimal search strategies for dynamic hypothesis testing.
     IEEE Transactions on Systems, Man and Cybernetics, 25 (July 1995), 1130–1138.

[6] D. A. Castañon
     Approximate dynamic programming for sensor management.
     In *Proceedings of the Thirty-sixth Conference on Decision and Control*, Dec. 1997, 1202–1207.

[7] S. Cheng, E. Leung, K. Lochner, K. O'Malley, D. Reeves, J. Schvartzman and M. Wellman
Walverine: A Walrasian trading agent.
Decision Support Systems, 39, 2 (Apr. 2005), 169–184.

[8] R. V. Denton, E. Alcaraz, J. Knopow, J. Llinas and K. Hintz
Towards modern sensor management systems.
In *Proceedings of the Sixth Joint Service Data Fusion Symposium*, June 14–18, 1993, 659–678.

[9] Y. Fujishima, K. Leyton-Brown and Y. Shoham
Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches.
In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (IJCAI), 548–553.

[10] A. V. Goldberg and J. D. Hartline
Envy-free auctions for digital goods.
In *Proceedings of the Fourth Association for Computing (ACM) Conference on Electronic Commerce* (EC-03), 29–35.

[11] D. L. Hall and J. Llinas
Handbook of Multisensor Data Fusion.
Boca Raton, FL: CRC Press LLC, 2001.

[12] D. L. Hall and S. McMullen
Mathematical Techniques in Multisensor Data Fusion (2nd ed.).
Norwood, MA: Artech House, 2004.

[13] K. J. Hintz and E. S. McVey
Multi-process constrained estimation.
IEEE Transaction on Systems, Man, and Cybernetics, 21, 1 (Jan./Feb. 1991), 434–442.

[14] K. J. Hintz and G. McIntyre
Goal lattices for sensor management.
In *Proceedings of the Signal Processing, Sensor Fusion and Target Recognition VIII: SPIE International Symposium on Aerospace/Defense Sensing and Control*, Apr. 1999, 249–255.

[15] K. J. Hintz and J. Malachowski
Dynamic goal instantiation in goal lattices for sensor management.
In *Proceedings of the SPIE*, May 2005, 93–99.

[16] J. S. Jordan
The competitive allocation process is informationally efficient uniquely.
Journal of Economic Theory, 28, 1 (1982), 1–18.

[17] M. Kalandros and L. Y. Pao
Covariance control for multisensor systems.
IEEE Transactions on Aerospace Electronic Systems, 38, 4 (Oct. 2002), 1138–1157.

[18] K. Kastella
Discrimination gain for sensor management in multitarget detection and tracking.
In *Proceedings of the IEEE—Systems, Man and Cybernetics (SMC)/International Association for Mathematics and Computers in Simulation (IMACS) Multi-Conference on Computational Engineering in Systems Applications* (CESA'96), July 1996, 167–172.

[19] J. D. Kattar
A solution of the multi-weapon, multi-target assignment problem.
WP-26597, The MITRE Corporation, Bedford, MA, 1986.

[20] M. Kolba and L. L. Collins
Information based sensor management in the presence of uncertainty.
IEEE Transactions on Signal Processing, 55, 6 (June 2007), 2731–2735.

[21] L.Walras
Elements of Pure Economics.
Homewood, IL: Irwin, 1954.

[22] V. Lesser, C. L. Ortiz and M. Tambe
Distributed Sensor Networks: A Multiagent Perspective.
Boston, MA: Kluwer Academic Publishers, 2003.

[23] G. Linden, B. Smith and J. York
Amazon.com recommendations: Item-to-item collaborative filtering.
IEEE Internet Computing, 7, 1 (Jan. 2003), 76–80.

[24] G. Mainland, D. C. Parkes and M. Welsh
Decentralized, adaptive resource allocation for sensor networks.
In *Proceedings of the Second The Advanced Computing Systems Association (USENIX)/Association for Computing Machinery (ACM) Symposium on Networked Systems Design & Implementation* (NSDI'05), Apr. 2005, pp. 7–13.

[25] G. A. McIntyre and K. J. Hintz
An information-theoretic approach to sensor scheduling.
In *Proceedings of the SPIE: Signal Processing, Sensor Fusion, and Target Recognition V*, Apr. 1996, 304–312.

[26] G. A. McIntyre and K. J. Hintz
Sensor management simulation and comparative study.
In *Proceedings of the SPIE International Symposium on Aerospace/Defense Sensing & Control: Signal Processing, Sensor Fusion, and Target Recognition VI*, Apr. 1997.

[27] J. Moffat
Complexity Theory and Network Centric Warfare.
CCRP Publication Series, 2003.

[28] T. Mullen, V. Avasarala and D. L. Hall
Bringing the market to level 4 fusion: Investigation of auction methods for sensor resource allocation.
In *Proceedings of the Military Sensing Symposia (MSS) National Symposium on Sensors and Data Fusion* (NSSDF), May 2005.

[29] T. Mullen, V. Avasarala and D. L. Hall
Customer-driven sensor management.
IEEE Intelligent Systems, 21, 2 (2006), 41–49.

[30] R. Muraleedharan and L. Osadciw
Decision in a building access system using swarm intelligence and posets.
In *Proceedings of the Conference on Information Sciences and Systems*, Mar. 2004, 414–419.

[31] J. M. Nash
Optimal allocation of tracking resources.
In *Proceedings of the IEEE Conference on Decision and Control*, Dec. 1977, 1177–1180.

[32] J. Ostwald, V. Lesser and S. Abdallah
Combinatorial auction for resource allocation in a distributed sensor network.
In *Proceedings of the The 26th IEEE International Real-Time Systems Symposium* (RTSS'05), Dec. 2005, pp. 266–274.

[33] P. Padhy, R. K. Dash, K. Martinez and N. R. Jennings
A utility-based sensing and communication model for a glacial sensor network.
In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS'06), May 2006, 1353–60.

[34] P. A. Samuelson
Foundations of Economic Analysis.
Cambridge, MA: Harvard University Press, 1947.

[35] T. Sandholm
Algorithm for optimal winner determination in combinatorial auctions.
Artificial Intelligence, 135, 1–2 (Feb. 2002), 1–54.

[36] W. Schmaedeke
Information-based sensor management.
In *Proceedings of the SPIE: Signal Processing, Sensor Fusion, and Target Recognition II*, Apr. 1993, 156–64.

[37] M. K. Schneider and C.-Y. Chong
A rollout algorithm to coordinate multiple sensor resources to track and discriminate targets.
In *Proceedings of the SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition*, Apr. 2006.

[38] J. B. Shoven and J. Whalley
Applying General Equilibrium.
New York: Cambridge University Press, 1992.

[39] H. R. Varian and J. K. MacKieMason
Generalized Vickrey auctions.
Department of Economics, University of Michigan, 1994.

[40] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart
and S. Stornetta
Spawn: A distributed computational economy.
IEEE Transactions on Software Engineering, 18, 103–117.

[41] W. E. Walsh, M. P. Wellman and F. Ygge
Combinatorial auctions for supply chain formation.
In *Proceedings of the Association for Computing Machinery
(ACM) Conference on Electronic Commerce* (EC'00), Oct.
2000, 260–269.

[42] W. E. Walsh, M. P. Wellman, P. R. Wurman and J. K. MacKie-
Mason
Some economics of market-based distributed scheduling.
In *Proceedings of the International Conference on Dis-
tributed Computing Systems*, May 1998, 612–621.

[43] R. Washburn, M. Schneider and J. Fox
Stochastic dynamic programming based approaches to sen-
sor resource management.
In *Proceedings of the Fifth International Conference on
Information Fusion*, 608–615.

[44] J. L. Williams, J. W. Fisher, III and A. S. Willsky
An approximate dynamic programming approach to a com-
munication constrained sensor management problem.
In *Proceedings of the Eighth International Conference on
Information Fusion*, July 2005.

[45] F. Ygge
Market-Oriented Programming and its Application to Power
Load Management.
Ph.D. thesis, Department of Computer Science, Lund Uni-
versity, ISBN 91-628-3055-4, CODEN LUNDFD6/(NFCS-
1012)/1-224/(1998).

Viswanath Avasarala obtained his bachelor's degree from the Indian Institute of Technology, Madras in 2001. He has a Ph.D. in information sciences and technology from Pennsylvania State University. In his Ph.D. thesis, he investigated the use of market-oriented multi-agent frameworks for real-time and distributed sensor management.

His primary research interests are in data mining, market-oriented programming and multi-agent collaboration. Since 2006, he has been working as an information scientist at GE Global Research. At GE, Dr. Avasarala uses artificial intelligence based methodologies to solve research problems in machinery diagnostics and prognostics, fleet logistics and health management, stochastic simulation, multi-objective optimization and medical imaging. He has published 3 journal papers, 1 book chapter and more than 10 conference papers in these areas. He has also chaired and/or acted as an organizing committee member of several reputed international conferences in soft computing. He is an active IEEE member and the current chair of Schenectady chapter of the IEEE Computational Intelligence Society.

Tracy Mullen is an assistant professor at the College of Information Sciences and Technology at The Pennsylvania State University. She received her Ph.D. in computer science from the University of Michigan.

She has previously worked at Lockheed Martin, Bellcore, and NEC Research. Dr. Mullen's research is focused on computational markets and multiagent systems to support decentralized decision making. Specific interests include market-based resource allocation for sensor management, prediction markets, and trading agents and markets. Her research papers have been published in *ACM Transactions on Internet Technology*, *Decision Support Systems*, *Electronic Commerce Research*, *IEEE Computer*, *Mathematics and Computers in Simulation*, and *Operating Systems Review*, among others.

David L. Hall is a professor in the Penn State College of Information Sciences and Technology (IST) where he leads the Center for Network Centric Cognition and Information Fusion. Previously he was the associate dean for Research and Graduate Programs in Penn State's College of Information Sciences and Technology, where he was responsible for overseeing IST's Ph.D. and Master's Degree programs as well as research grant administration and leadership of IST faculty research efforts. Hall joined IST from the University's Applied Research Laboratory (ARL) where he served as associate director and senior scientist. At ARL, he oversaw the 150-person Information and Network Systems Office, comprised of four divisions: Information Science and Technology, Navigation Research and Development, Systems and Operations Automation, and Communications Science and Technology. He has more than 25 years of experience in research, research management, and systems development.

He is the author of more than 200 papers, reports, books and book chapters, and he has delivered numerous lectures on his research, research management, and artificial intelligence. Hall has been named as an IEEE fellow for his contributions to data fusion and he was awarded the DoD Joe Mignona National Data Fusion award for his contributions and leadership in data fusion.