Distributed Sensor Allocation for Multi-Target Tracking in Wireless Sensor Networks

YINFEI FU
Michigan Technological University

QING LING, Member, IEEE
University of Science and Technology of China

ZHI TIAN, Senior Member, IEEE
Michigan Technological University

In emerging tracking systems using large-scale wireless sensor networks, sensor management is an essential task in order to balance the tracking performance and costs subject to limited network resources in terms of energy, communication bandwidth, and sensing range. This paper considers the sensor allocation problem for multi-target tracking (MTT), in which a group of sensors are dynamically selected and allocated to track each of the multiple targets and collaborate within the group via track data fusion. The sensor assignments evolve over time as targets move, and are accomplished by solving a constrained optimization problem that is formulated to maximize the overall tracking performance for all targets, while conserving network energy and providing tracking coverage guarantee. The original integer-valued optimization problem is relaxed to a convex program for computational tractability, and then implemented in a distributed manner for network scalability and reduced communication costs. Through local one-hop communication with neighboring nodes, each sensor autonomously decides on whether to participate in data collection and how to contribute to track fusion. The proposed distributed sensor allocation algorithm, implemented via iterative subgradient search, is shown to converge to the global optimum of the centralized relaxed problem, and is near optimal for the original integer programming problem.

Manuscript received June 21, 2010; revised February 11 and September 19, 2011; released for publication February 7, 2012.

IEEE Log No. T-AES/48/4/944220.

Refereeing of this contribution was handled by S. Coraluppi.

Y. Fu and Z. Tian are supported in part by the U.S. National Science Foundation (ECS-0925881 and CNS-1017887). Q. Ling is supported in part by National Natural Science Foundation of China (61004137), National High-tech R&D Program of China (863 Program, 2009AA01A340), and Fundamental Research Funds for the Central Universities.

Authors' addresses: Y. Fu and Z. Tian, Department of Electrical & Computer Engineering, Michigan Technological University, Houghton, MI 49931; Q. Ling, Department of Automation, University of Science and Technology of China (West Campus), Hefei, Anhui, China, 230027, E-mails: (qingling@mail.ustc.edu.cn) with cc to (ztian@mtu.edu).

I. INTRODUCTION

Wireless sensor networks have become increasingly important in a number of civilian and military applications. Sensor nodes can be densely deployed in a large sensing field to acquire information about the physical phenomena of interest. Densely scattered nodes provide overlapping coverage, which enhances robustness and improves sensing quality and accuracy. This paper concerns the problem of target tracking over a large field using distributed wireless sensor networks, which is of interest to a number of applications including security and surveillance, battlefield information monitoring, traffic management, wild animal tracking, and environmental monitoring [1–5].

The goal of tracking is to obtain trajectories of targets moving over the sensing field. Traditional tracking systems are radar based, for which the issues of data association and track fusion have been widely studied [1, 4, 6–8]. Emerging tracking systems feature in networks of a large number of small sensors that can be densely deployed over the monitoring field. The new paradigm results in new challenges due to bandwidth and energy limitations of the network. Meanwhile, the limited sensing coverage of each sensor suggests that it is infeasible for a target to be tracked by a fixed subset of static sensors over the entire moving range. Further, given a large sensing field and the localized nature of targets, it is highly ineffective to invoke all sensors in the entire tracking period, since sensors far away from the targets do not contribute much to the tracking quality, but consume power and bandwidth in collecting data and communicating within the large network. These issues give rise to the sensor management problem, which aims to assign a sequence of sensors to track a moving target. When there are multiple moving targets, different sets of sensors need to be scheduled to cooperatively track individual targets at different instants of the time horizon, due to the limited sensing range, transmission power and tracking capability of a single sensor. As a result, cooperation among sensor nodes becomes necessary yet challenging. Each sensor needs to trade off the options of whether to track, which target to track, and how to respond to the dynamic behavior of different targets. Cooperating sensors also need to perform proper track fusion in real time, at the expense of data communication that consumes network resources. These sensor management issues are not considered in traditional tracking systems. Overall, the goal is to balance the tracking quality against system resource constraints based on certain performance criterion, particularly, to determine the best way to select a node or a group of nodes to gain information on each of the targets.

For sensor scheduling, most work so far focuses on the single target case, see, e.g., [9–13]. An information-driven sensor querying and data routing approach is developed in [12], which activates only a single sensor (a.k.a. leader) at any given time. The leader selects the best sensor node in the network by optimizing either a Mahalanobis distance or an entropy measure from the current track estimate and the sensor position, and passes the measurement to this node which becomes the new leader. This is a heuristic approach that selects the sensors whose Euclidean distances to the target are less than a threshold. It confines to the simple case of selecting only one sensor node at each time slot for the tracking task, and is mainly suited for ranging nodes. A sensor selection scheme based on the Kullback-Leibler divergence is discussed in [13], under the assumption that no more than one target can appear simultaneously within the sensing range of any single sensor.

For the case of multi-target tracking (MTT), the management of sensor resources in wireless sensor networks has attracted increasing attention recently [8, 14–17]. Sensor management is intimately related to track fusion, where multiple sensors collaboratively track the same target in order to improve tracking performance. Therefore it is important to allocate proper nodes to track the proper targets at each time slot, in order to attain the best overall tracking performance. Such a task is challenging for the MTT case, with few works in the literature. For centralized global sensor allocation, a covariance control strategy is developed in [18], [19], which uses a heuristic search to select sensor combinations based on the difference between the desired error covariance matrix and the predicted covariance of each target. Information theoretic principles for global sensor management are discussed in [20]–[22].

This paper focuses on the distributed sensor allocation problem for the MTT case. Not only do we have to perform sensor selection for multiple targets, but also we allow multiple sensors in each time slot to be active in tracking the same target via track fusion. This scenario is more practical and can result in higher tracking performance, compared with the simple case of scheduling a single sensor for tracking a single target. Further, we allow multiple targets to appear simultaneously within the coverage of a sensor node, so that each sensor needs to decide not only whether to track, but also which target to track. To the best of our knowledge, such scenarios have not been investigated in the literature.

Sensor management for MTT also concerns the well-known data association issue, which by itself has been extensively investigated for years. Some prominent techniques include multiple hypothesis tracking (MHT), joint probabilistic data association filter (JPDAF) and the joint multi-target probability density (JMPD) method [1, 7, 8], to name a few. This paper focuses on the emerging sensor management problem for wireless sensor networks, and assumes that data association has been perfectly accomplished. This assumption is made to avoid diverting much into those common design issues in traditional MTT systems, because the impact of data association errors eventually needs to be evaluated in terms of the tracking performance degradation.

Considering the network resource constraints in large-scale wireless sensor networks, this paper develops decentralized sensor management algorithms. Most existing sensor allocation schemes for MTT are essentially centralized, which require access to global information in order to reach globally optimal solutions [18, 19]. Unfortunately, the sensor allocation problem is an integer program by nature, which is computationally challenging for a large-scale network because the search space can be as large as the number of sensors in the network. A direct approach to solving the sensor allocation problem is based on exhaustive search over all sensor-target assignment options [24], which causes combinatorial complexity and heavy communication load as well. Other techniques applied to solving this problem include branch and bound [10], greedy search [3, 18, 23, 24] and heuristic search [19]. These methods are either computationally complex or suboptimal in performance. To reduce the complexity, an approximate convex optimization formulation is presented that bypasses integer programming via linear programming relaxation [25]. Nevertheless, it does not consider data fusion among multiple sensors, nor does it treat the multiple-target case; also, it presents the centralized formulation without concerning scalable and distributed implementations.

Evidently, for a large-scale wireless sensor network, it is crucial to develop efficient and robust sensor allocation algorithms that are scalable in computing and communication costs. This paper relaxes the original integer-valued allocation problems into convex optimization formulations, and then implements the solutions in a distributed manner using the primal-dual approach and iterative subgradient search with guaranteed convergence. Only local information exchange is needed among one-hop neighboring nodes. Through iterations, the developed distributed algorithms attain global optimality of the relaxed centralized formulations. Because of the small performance gap caused by the convex relaxation to the centralized optimization formulations, our distributed sensor allocation framework offers scalable and near-optimal solutions to the original integer programming problems.

The remainder of this paper is organized as follows. Section II provides a system framework for multi-sensor MTT. The sensor allocation problem in an MTT system is formulated as an integer-valued optimization problem in Section III. The solution to this problem is discussed in Section IV, where novel distributed sensor allocation algorithms are developed for MTT. Simulation results are presented in Section V, followed by summarizing remarks in Section VI.

II. SYSTEM ARCHITECTURE

Consider the problem of tracking multiple moving targets over a large area monitored by a network of wireless sensors. Sensors are deployed densely enough such that the entire area is under the coverage of the network, whereas each sensor only has a limited sensing range. Each sensor knows its own position via network calibration during the deployment stage, but to save memory costs, does not necessarily know the positions of other sensors. During the tracking stage, it is possible that a target is within the sensing range of multiple sensors at a time. When this situation arises, we allow multiple sensors to track the same target and collaborate via track data fusion. Meanwhile, we assume that each sensor has the capability to sense and track multiple targets within the same time window, when these targets move within the sensing range of this sensor. Nevertheless, a sensor may decide not to track a target within its own sensing area, under the tradeoff consideration of tracking performance versus sensing costs. Further, when a target moves, it cannot be tracked by the same group of sensors all the time, and hence sensors have to dynamically decide whether and when to join or leave the tracking task. These issues give rise to the sensor management problem of interest in this paper, where our objective is to dynamically allocate a group of sensor nodes to track each of the multiple targets, in order to optimize the overall tracking performance under certain network resource constraints. In particular, we focus on the case of distributed tracking and sensor management, in the absence of a fusion center.

For MTT we adopt a multi-layer architecture for the distributed tracking system, in which multiple modules are intertwined in a hierarchical manner, with each module focusing on one designated task [3, 6]. These modules are: sensing, sensor management, target measurement collection, multi-target tracking, and information relay. Specifically, at the physical-layer sensing module, sensors detect targets within its sensing range by measuring the received signal-to-noise ratio (SNR) of targets and report these SNR values to the upper-level module for action decision. The sensor allocation module dynamically allocates a group of sensors to track each target within each time window. As soon as a node is assigned to track a target, the target measurement collection module of each selected node is triggered to collect position-related measurements, typically in the form of time-of-arrival (TOA) and/or direction-of-arrival (DOA) data. Afterward, the tracking and fusion module takes place, in which a tracking algorithm, say sequential Kalman filtering, is carried out by a group of nodes to update the states of target positions and velocities. Finally, when a target moves from the sensing range of one group of sensors to that of the next group, necessary information needs to be passed on through the information relay module, in order to ensure uninterrupted sensing and track update. Overall, the sensing module is active for all sensors, while the other modules are activated only for selected sensors on an as-needed basis, in order to conserve network energy. Detailed implementation of this multi-layer architecture for the proposed MTT system is elaborated in Table I in Section IV-B.

This paper focuses on the sensor management module, which has received limited treatment for the MTT problem, especially in the distributed scenarios. For other modules, there has been a large body of literature given the traditional results and recent development [1, 6, 8, 12, 13, 26, 27]. Hence, whenever functionalities from other modules are called for, we refer to the literature. Several remarks are in order regarding these MTT modules and the basic assumptions made.

A1) A distinct design choice in our tracking architecture is the separation of the sensing module based on SNR measurements and the target measurement collection module based on TOA/DOA data. Wireless sensor networks face stringent energy constraints and have to judiciously reduce energy consumption in order to extend the network lifetime. Therefore, forcing all sensor nodes to do active sensing is energy-consuming. An energy-saving strategy is to put some sensors to sleep, but this cannot be applied straightforwardly in tracking applications, because of the unacceptable impact of loss of tracks. We adopt a two-step approach, such that most sensor nodes only need to operate in the simple sensing module by measuring SNRs whereas only a small group of properly selected sensors need to perform active sensing by activating their TOA/DOA measurement module at a given time; this approach significantly reduces the energy consumption of active sensing.

A2) It is assumed that data association for sensor management has been performed to link sensed measurements with corresponding targets. This can be done using techniques in the literature, possibly using joint target identification and tracking [3, 4]. Details of data association and impact of association errors are out of the scope of this paper. Note that sensor management concerns making sensor-target allocation decisions to save network resources, which have not touched upon tracking decisions yet. In the ensuing tracking module, the association errors might be corrected or might propagate, depending on the joint data association and tracking algorithm used and the specific scenario under investigation. Therefore, even when the allocation decisions are affected by imperfect data association, the impact is mainly on the network energy efficiency.

A3) For each sensor, the communication range r_C is set to be at least twice of the sensing range r_S, which is a necessary condition for the operations in the information relay module. Nevertheless, r_C may still be quite small compared with the large sensing area, such that all the communications are localized.

Next, we provide the signal models for those modules relevant to sensor management.

A. Data Model

Suppose that there are a total of N_s sensors and N_a targets, with $N_a \geq 1$. The two-dimensional position of the ith sensor is assumed to be known to itself, denoted by $\mathbf{p}_{s,i} = (x_{s,i}, y_{s,i})$, $i = 1, \ldots, N_s$, while that of the jth target is unknown, denoted by $\mathbf{p}_{a,j} = (x_{a,j}, y_{a,j})$, $\forall j = 1, \ldots, N_a$. In the sensing module sensors measure the received signal strength (RSS) and collect the SNR data which depend on the square-distances $d_{ij}^2 = \|\mathbf{p}_{s,i} - \mathbf{p}_{a,j}\|_2^2$, $\forall i, j$. It is assumed that a sensor i can detect a target j within its sensing range, when the corresponding distance d_{ij} is smaller than the sensing range r_S. Because the RSS can be measured at relatively low costs, sensors can afford to keep their sensing module on, which is essential to avoid loss of track and loss of optimality in sensor management.

In the track measurement collection module, selected sensors collect proper measurements to calibrate the positions of their allocated targets. This can be done by collecting range-only measurements such as TOA and time-difference-of-arrival (TDOA), or bearing-only measurements such as DOA. When a single measurement type is collected, multiple sensor nodes need to be involved to extract full position information of a single target in the tracking process, using for example, multi-node triangularization techniques. Several sensor allocation schemes based on bearing-only parameters have been developed [24]. This practice reduces the costs of individual sensors, at the expense of an increased number of nodes needed for triangularization, higher communication bandwidth consumption for extensive information exchange, and/or network-wide time synchronization. These network-level costs can be alleviated and traded off with increased sensor costs using a joint TOA/DOA approach [27, 28], which we adopt in this paper. In our work we suppose that each sensor node can calibrate both the range and bearing of a target to make individual position estimation. Owing to the two-step sensing approach, a sensor performs the joint TOA/DOA-based positioning only when it is allocated to join the tracking task for a target. In the case of noncooperative target tracking, a selected sensor switches to the active sensing mode, and computes the TOA based on the round trip delay from its transmitter to the target and back. When not selected, a sensor turns off its track measurement collection module to conserve energy.

Let $\hat{\tau}_{ij}$ and $\hat{\theta}_{ij}$ denote the TOA and DOA data of the jth target measured by the ith sensor. The true values of τ_{ij} and θ_{ij} are given by $\tau_{ij} = d_{ij}/c$ and $\theta_{ij} = \tan^{-1}((y_{a,j} - y_{s,i})/(x_{a,i} - x_{s,i}))$, respectively, where c is the propagation speed, which could be the speed of light or the speed of sound depending on the modality of sensor. Hence, the data model is given by

$$\begin{aligned} \hat{\tau}_{ij} &= \tau_{ij} + \Delta\tau_{ij} \\ \hat{\theta}_{ij} &= \theta_{ij} + \Delta\theta_{ij} \end{aligned} \tag{1}$$

where $\Delta\tau_{ij}$ and $\Delta\theta_{ij}$ are additive data noises induced by estimation errors, subject to interference from other targets. It is assumed that the noises are white Guassian with distributions $\Delta\tau_{ij} \sim \mathcal{N}(0, \sigma_{ij})$ and $\Delta\theta_{ij} \sim \mathcal{N}(0, \eta_{ij})$, where the variances σ_{ij} and η_{ij} depend on the unbiased TOA and DOA estimators adopted [29].

Using the TOA and DOA data, the target position $\mathbf{p}_{a,j}$ can be estimated by sensor i as follows:

$$\begin{aligned} \hat{x}_{ij} &= x_{s,i} + c\hat{\tau}_{ij}\cos(\hat{\theta}_{ij}) \\ \hat{y}_{ij} &= y_{s,i} + c\hat{\tau}_{ij}\sin(\hat{\theta}_{ij}). \end{aligned} \tag{2}$$

With (1) and (2), the position estimates $(\hat{x}_{ij}, \hat{y}_{ij})$ can be described as

$$\begin{aligned} \hat{x}_{ij} &= x_{i,j} + \Delta x_{ij} \\ \hat{y}_{ij} &= y_{i,j} + \Delta y_{ij} \end{aligned} \tag{3}$$

where the position errors Δx_{ij} and Δy_{ij} are Gaussian random variables with zero mean and variance of $\delta^2_{\Delta x_{ij}}$ and $\delta^2_{\Delta y_{ij}}$, respectively. The Cramer-Rao bounds (CRBs) of $\delta^2_{\Delta x_{ij}}$ and $\delta^2_{\Delta y_{ij}}$ are derived in [29].

B. Multi-target Tracking Model

The track state vector of target j is defined as $\mathbf{u}_j = [x_{a,j}, y_{a,j}, \dot{x}_{a,j}, \dot{y}_{a,j}]^{\mathrm{T}}$, where $\dot{x}_{a,j}$ and $\dot{y}_{a,j}$ are the target velocity projected onto the x and y coordinates, respectively. At each time slot t, the target motion obeys a linear discrete-time Markov process, modeled as

$$\mathbf{u}_j(t) = \mathbf{A}_j(t)\mathbf{u}_j(t-1) + \mathbf{C}_j(t)\mathbf{v}_j(t) \tag{4}$$

where $\mathbf{A}_j(t)$ is the state transition matrix and $\mathbf{C}_j(t)\mathbf{v}_j(t)$ is process noise with $\mathbf{v}_j(t)$ following Gaussian distribution, i.e., $\mathbf{v}_j(t) \sim N(0, \mathbf{Q}_j(t))$.

At each sensor i, its measurement vector $\mathbf{z}_{ij}$ for the target j can be modeled as

$$\mathbf{z}_{ij}(t) = \mathbf{H}_{ij}(t)\mathbf{u}_j(t) + \mathbf{w}_{ij}(t) \tag{5}$$

where $\mathbf{H}_{ij}(t)$ is the measurement matrix, and $\mathbf{w}_{ij}(t)$ is the measurement noise of Gaussian distribution $\mathbf{w}_{ij}(t) \sim N(0, \mathbf{R}_{ij}(t))$. If the data model (3) is adopted, then $\mathbf{z}_{ij}(t) = [\hat{x}_{ij}, \hat{y}_{ij}]^{\mathrm{T}}$ and $\mathbf{H}_{ij}(t)$ is simplified to

$$\mathbf{H}_{ij}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \qquad \forall i, j, t.$$

The sensor allocation algorithm developed in this paper focuses on this commonly used case, but is directly applicable to the general case of any $\mathbf{H}_{ij}$.

In the model set (4) and (5), the process noise covariance matrix $\mathbf{Q}_j(t)$ depends on the target motion and is typically assumed to be time invariant and known a priori. The measurement noise covariance matrix $\mathbf{R}_{ij}(t)$, on the other hand, depends on the received SNR as well as the relative positions of the target and sensors, and hence varies over time as the target move. In the tracking model discussed in this section, we assume that variance of the measurement noise changes as the observation environment varies, which fits the practical scenario but is rarely considered in existing literature. Indeed, in a practical tracking process, the acquisition accuracy of position-dependent parameters such as TOA, DOA, or RSS depends closely on factors in the wireless channel environment, including the instantaneous SNR values and target-sensor distances that change dynamically for mobile targets. Because it is difficult to acquire information of the variance of measurement noise in a real-time environment, we use instead the analyzed CRB of the measurement noise as the prediction of the noise covariance in our sensor allocation algorithm, which is elaborated later. Particularly, the CRB of $\mathbf{R}_{ij}$, denoted as $\mathbf{M}_{ij}$, has been derived in [29], which is used in lieu of $\mathbf{R}_{ij}$ in guiding the decision making in our sensor management module. Through sensor management, the network autonomously allocates a group of sensors $S_j(t)$ to track the target j at time t, $\forall j$, and dynamically updates the members of $S_j(t)$ as the target moves over time.

C. Sequential Kalman Filter for Track Fusion

After target-sensor allocation decisions $\{S_j(t)\}_{j=1}^{N_a}$ have been made in the sensor management module, track fusion for each target j is performed via cooperation among all sensors in $S_j(t)$, $\forall j$. To do so, Kalman filtering (KF) can be applied based on the linear dynamic model and the measurement model in (4) and (5).

Following the notational convention of KF, we let $\hat{\mathbf{u}}_j(t \mid t-1)$ denote the predicted a priori state vector of target j at time t given all the available measurements up to $t-1$, and $\hat{\mathbf{u}}_j(t \mid t)$ denote the updated a posteriori state estimate after incorporating the measurements $\{\mathbf{z}_{ij}(t)\}_i$ from the allocated sensors $i \in S_j(t)$. Correspondingly, the prior and posterior estimation errors for target j are $\mathbf{e}_j(t \mid t-1) = \mathbf{u}_j(t) - \hat{\mathbf{u}}_j(t \mid t-1)$ and $\mathbf{e}_j(t \mid t) = \mathbf{u}_j(t) - \hat{\mathbf{u}}_j(t \mid t)$, respectively, whose error covariance matrices are defined as $\mathbf{P}_j(t \mid t-1) = E[\mathbf{e}_j(t \mid t-1)\mathbf{e}_j^{\mathrm{T}}(t \mid t-1)]$ and $\mathbf{P}_j(t \mid t) = E[\mathbf{e}_j(t \mid t)\mathbf{e}_j^{\mathrm{T}}(t \mid t)]$, where $(\cdot)^{\mathrm{T}}$ denotes transpose and $E[\cdot]$ denotes expectation.

To facilitate sensor management, we adopt an information form of KF for track fusion, in which the state vector estimate $\hat{\mathbf{u}}_j$ and the estimate error covariance matrix $\mathbf{P}_j$ are transformed into the information state vector $\hat{\mathbf{y}}_j = \mathbf{P}_j^{-1}\hat{\mathbf{u}}_j$ and the information matrix $\mathbf{Y}_j = \mathbf{P}_j^{-1}$, respectively [6, 30]. Evidently, the updates $(\hat{\mathbf{u}}_j, \mathbf{P}_j)$ in the traditional KF can be equivalently obtained from $(\hat{\mathbf{y}}_j, \mathbf{Y}_j)$, which are given by the transformed KF updating rule, as follows [6, 30]:

Prediction:

$$\hat{\mathbf{y}}_j(t \mid t-1) = \mathbf{Y}_j(t \mid t-1)\mathbf{A}_j(t)\mathbf{Y}_j^{-1}(t-1 \mid t-1)\hat{\mathbf{y}}_j(t-1 \mid t-1) \tag{6a}$$

$$\mathbf{Y}_j^{-1}(t \mid t-1) = \mathbf{A}(t)\mathbf{Y}_j^{-1}(t-1 \mid t-1)\mathbf{A}^{\mathrm{T}}(t) + \mathbf{C}(t)\mathbf{Q}(t)\mathbf{C}^{\mathrm{T}}(t) \tag{6b}$$

Estimation:

$$\hat{\mathbf{y}}_j(t \mid t) = \hat{\mathbf{y}}_j(t \mid t-1) + \sum_{i \in S_j(t)} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{R}_{ij}^{-1}(t)\mathbf{z}_{ij}(t) \tag{6c}$$

$$\mathbf{Y}_j(t \mid t) = \mathbf{Y}_j(t \mid t-1) + \sum_{i \in S_j(t)} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{R}_{ij}^{-1}(t)\mathbf{H}_{ij}(t). \tag{6d}$$

The above information form of KF is equivalent to the traditional KF in terms of tracking performance, and provides computational advantages for multi-sensor data fusion [6, 30]. Since the sensor measurements contribute in an additive manner in (6c) and (6d), it is simple to compute and amenable to distributed implementation, e.g., using sequential KF [6]. Furthermore, the information matrix $\mathbf{Y}_j$ is associated with the Fisher information defining the CRB of the target state estimate, which is utilized in the sensor allocation optimization problem presented in the next section.

III. PROBLEM FORMULATION FOR SENSOR MANAGEMENT

This section provides the optimization formulation for the sensor management problem in MTT systems. The goal is to optimize the expected overall tracking performance from the information theoretic view of the KF.

A. Characterization of Tracking Performance

The tracking performance is dictated by the uncertainty of the target state, which can be quantitatively measured by entropy-based information-theoretic metrics [20, 21]. The following Proposition 1 reveals a useful relationship between the entropy of a target state $\mathbf{u}_j(t)$ in (4) and the corresponding information matrix $\mathbf{Y}_j(t \mid t)$ in (6d) that can be acquired from practical KF operations.

PROPOSITION 1 *Given the linear Markov Gaussian dynamic model* (4) *and the measurement model* (5)*, the conditional entropy* $H[\cdot \mid \cdot]$ *of the state of the target* j *given all the observations up to* t *is given by*

$$H[\mathbf{u}_j(t) \mid Z_j(t)] = \gamma - \tfrac{1}{2}\log(|\mathbf{Y}_j(t \mid t)|) \tag{7}$$

where γ *is a constant,* $|\cdot|$ *denotes the matrix determinant, and* $Z_j(t)$ *is the set of available observations up to time* t *for target* j*, i.e.,* $Z_j(t) = \{\bar{\mathbf{z}}_j(0), \bar{\mathbf{z}}_j(1), \ldots, \bar{\mathbf{z}}_j(t)\}$ *with* $\bar{\mathbf{z}}_j(t') = \{\mathbf{z}_{ij}(t')\}_{i \in S_j(t')}$*,* $t' \in [0, t]$.

PROOF Consider the target j that is tracked from measurements $Z_j(t)$ over time. Because the state vector $\mathbf{u}_j(t)$ is described by a linear Markov Gaussian dynamic model, the KF can be derived based on the Bayesian framework, which leads to the Gaussian distribution $p(\mathbf{u}_j(t) \mid Z_j(t)) \sim \mathcal{N}(\hat{\mathbf{u}}_j(t \mid t), \mathbf{P}_j(t \mid t))$. Following the entropy definition $H[\mathbf{u}_j(t) \mid Z_j(t)] = -E[\log p(\mathbf{u}_j(t) \mid Z_j(t))]$, and noting that the KF is unbiased with $E[\hat{\mathbf{u}}_j(t \mid t)] = \mathbf{u}_j(t)$, it can be shown that (7) holds.

Since the KF estimate of $\mathbf{u}_j(t)$ is unbiased, efficient and consistent, the corresponding Fisher information matrix $\mathbf{T}_j(t)$ is equal to the inverse of the posterior error covariance matrix $\mathbf{P}_j(t \mid t)$ that reaches the CRB [32], that is,

$$\begin{aligned}\mathbf{T}_j(t) &= \mathbf{Y}_j(t \mid t)\\ &= \mathbf{Y}_j(t \mid t-1) + \sum_{i \in S_j(t)} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{R}_{ij}^{-1}(t)\mathbf{H}_{ij}(t).\end{aligned} \tag{8}$$

Hence, the entropy of the target j is inversely proportional to the determinant of its CRB that quantifies the volume of the confidence ellipsoid for the target state estimation errors [6]. The results in (7) and (8) connect the mutual information from the information theoretic view to the CRB metric from the estimation theoretic view. With the equivalence, our goal of maximizing the tracking performance boils down to minimizing the uncertainty of the target state via proper management of the intertwined sensor allocation and data collection problems.

B. Optimization Formulation

The sensor allocation problem amounts to deciding the sensor set $S_j(t)$ for each target j at time t under system constraints, such that the collected measurements $\{\mathbf{z}_{ij}(t)\}_{i \in S_j(t)}$, along with past data $Z_j(t-1)$, maximally reduce the target state uncertainty quantified by (7). Since all the N_a targets are independent, the network-wide sum uncertainty is given by $\sum_{j=1}^{N_a} H[\mathbf{u}_j(t) \mid Z_j(t)]$ at time t, which is to be minimized.

Noting that $Z_j(t) = \{Z_j(t-1), \bar{\mathbf{z}}_j(t)\}$, we can write

$$H[\mathbf{u}_j(t) \mid Z_j(t)] = H[\mathbf{u}_j(t) \mid Z_j(t-1)] - I[\mathbf{u}_j(t) \mid \bar{\mathbf{z}}_j(t)]$$

where $I[\mathbf{u}_j(t) \mid \bar{\mathbf{z}}_j(t)]$ is the mutual information gain contributed from $\bar{\mathbf{z}}_j(t) = \{\mathbf{z}_{ij}(t), \forall i \in S_j(t)\}$, and similar to (7), the prediction uncertainty is given by $H[\mathbf{u}_j(t) \mid Z_j(t-1)] = \gamma - \frac{1}{2}\log(|\mathbf{Y}_j(t \mid t-1)|)$, $\forall j$, which is not dependent on $S_j(t)$. Let us assume that the network obtains the maximal information gain for all targets at time $(t-1)$. Hence, as far as $S_j(t)$ is concerned, minimizing $\sum_j H[\mathbf{u}_j(t) \mid Z_j(t)]$ is equivalent to maximizing the sum information gain $\sum_j I[\mathbf{u}_j(t) \mid \bar{\mathbf{z}}_j(t)]$, where $I[\mathbf{u}_j(t) \mid \bar{\mathbf{z}}_j(t)] = -H[\mathbf{u}_j(t) \mid Z_j(t)] + H[\mathbf{u}_j(t) \mid Z_j(t-1)]$ can be derived from (7) as

$$\begin{aligned}&I[\mathbf{u}_j(t) \mid \bar{\mathbf{z}}_j(t)]\\ &\quad = \frac{1}{2}\log\left(\left|\mathbf{I} + \mathbf{P}_j(t \mid t-1)\sum_{i \in S_j(t)} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{R}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)\right|\right).\end{aligned} \tag{9}$$

Thus, the network objective of minimizing the network sum uncertainty $\sum_j H[\mathbf{u}_j(t) \mid Z_j(t)]$ becomes

$$\min_{\{S_j(t)\}_{j=1}^{N_a}} -\sum_{j=1}^{N_a}\log\left(\left|\mathbf{I} + \mathbf{P}_j(t \mid t-1)\sum_{i \in S_j(t)} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)\right|\right). \tag{10}$$

In (10) we have used the CRB $\mathbf{M}_{ij}$ of the position measurement errors to replace the corresponding error covariance matrix $\mathbf{R}_{ij}$, $\forall i, j$, because our position measurements can be obtained from raw TOA and DOA data using efficient estimators that reach the CRB. Note that the CRB expressions for $\mathbf{M}_{ij}$ as functions of the actual target-node geometry $\{(\tau_{ij}, \theta_{ij})\}_{i,j}$ are known [29]. In practice $\mathbf{M}_{ij}(t)$ is replaced by its predicted value $\hat{\mathbf{M}}_{ij}(t \mid t-1)$ as a function of the predicted target-node geometry from the position estimates $\hat{\mathbf{u}}_j(t-1 \mid t-1) = \mathbf{Y}_j^{-1}(t-1 \mid t-1)\hat{\mathbf{y}}_j(t-1 \mid t-1)$ obtained from (6) at time $(t-1)$.

The set selection problem in (10) can be transformed into a member assignment problem by introducing the binary allocation variables $a_{ij}(t) \in \{0, 1\}$, $\forall i, j$, which takes the value 1 if $i \in S_j(t)$ and 0 otherwise. As such, the optimization over the sets $\{S_j(t)\}$ can be equivalently performed by optimizing over $\{a_{ij}(t)\}_{i,j}$.

The design of $\{a_{ij}(t)\}_{i,j}$ is subject to several constraints arising from sensor capability limitations and network energy consumption considerations. First, a sensor i can track a target j only when the target falls within the sensing range, that is, $d_{ij} \le r_S$. Let $\Pi_i(t)$ denote the set of targets within the sensing range of the sensor i at time t, and $\Omega_j(t)$ the set of sensors that are close enough to detect the target j at time t. Accordingly, the total number of elements in these sets are $|\Pi_i(t)|$ and $|\Omega_j(t)|$, respectively. To avoid the loss of track, it is required that each target be tracked by at least one sensor, with $|\Omega_j(t)| \geq 1$. Given the global constraint of covering all targets in the sensing area, we postulate the following requirement:

$$\sum_{i\in\Omega_j(t)} a_{ij}(t) \geq 1, \qquad \forall j = 1,\ldots,N_a. \tag{11}$$

Next, the tracking capability of a sensor is limited by its processing ability, memory size, and energy supply. For example, although we allow a sensor to collect measurements from multiple targets, a sensor can only process targets one by one. Meanwhile, to ensure successful tracking of a dynamic trajectory, the time interval between two adjacent sampling instants $(t-1)$ and t has to be short enough. Given a limited processing time and memory constraints, the sensor i may only process up to N_i targets at each sampling time with $N_i \geq 1$. The limited tracking capability of each sensor can be expressed as

$$\sum_{j\in\Pi_i(t)} a_{ij}(t) \leq N_i, \qquad \forall i = 1,\ldots,N_s. \tag{12}$$

Putting together (10), (11), and (12), we have formulated an optimization problem for sensor selection:

$$\max_{\{a_{ij}(t)\}_{i,j}} \quad \sum_{j=1}^{N_a} J_j(t) \tag{13a}$$

$$\text{s.t.} \quad \sum_{i\in\Omega_j(t)} a_{ij}(t) \geq 1, \qquad \forall j = 1,\ldots,N_a \tag{13b}$$

$$\sum_{j\in\Pi_i(t)} a_{ij}(t) \leq N_i, \qquad \forall i = 1,\ldots,N_s \tag{13c}$$

$$a_{ij} \in \{0,1\}, \qquad \forall i = 1,\ldots,N_s, \quad j = 1,\ldots,N_a \tag{13d}$$

where $J_j(t) = \log(|\mathbf{I} + \mathbf{P}_j(t \mid t-1)\sum_{i=1}^{N_s} a_{ij}\mathbf{H}_{ij}^{\mathrm{T}}(t) \cdot\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)|)$, $\forall j$, as in (10). Since the objective function seeks to maximize a performance metric without cost penalty, (13c) degenerates to an equality constraint as long as $|\Pi_i(t)| \geq N_i$. We use this inequality constraint for generality, because N_i is set prior to acquiring $|\Pi_i(t)|$.

The objective function in (13a) is a concave function of $\{a_{ij}\}$. To see this, we can rewrite $J_j(t)$ as $\log(|\mathbf{I} + \mathbf{P}_j^{1/2}(t \mid t-1)\sum_{i=1}^{N_s} a_{ij}\mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t) \cdot\mathbf{P}_j^{1/2}(t \mid t-1)|)$. This log-determinant function is concave since $\mathbf{P}_j^{1/2}(t \mid t-1)\sum_{i=1}^{N_s} \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t) \cdot\mathbf{P}_j^{1/2}(t \mid t-1)$ is a positive semi-definite matrix [34]. Meanwhile, (13b) and (13c) are linear inequality constraints. Due to the binary constraint in (13d), (13) can be classified as a Boolean convex optimization problem.

The constraint (13b) implies that data fusion is applied when a target is tracked by more than one sensor node. A special case arises when each target is tracked by only one sensor and each sensor can only track one target at each time t with $N_i = 1$, $\forall i$. When there is only one target, namely $N_a = 1$, it further reduces to the single sensor selection problem considered in [12]. In the case of $N_i = 1$, (13) can be transformed into an integer linear programming problem, as follows:

$$\max_{\{a_{ij}(t)\}_{i,j}} \quad \sum_{j=1}^{N_a}\sum_{i=1}^{N_s} a_{ij}(t)\log(|\mathbf{I} + \mathbf{P}_j(t \mid t-1) \times \mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)|) \tag{14a}$$

$$\text{s.t.} \quad \sum_{i\in\Omega_j(t)} a_{ij}(t) = 1, \qquad \forall j = 1,\ldots,N_a \tag{14b}$$

$$\sum_{j\in\Pi_i(t)} a_{ij}(t) \leq 1, \qquad \forall i = 1,\ldots,N_s \tag{14c}$$

$$a_{ij} \in \{0,1\}, \qquad \forall i = 1,\ldots,N_s, \quad j = 1,\ldots,N_a. \tag{14d}$$

IV. DISTRIBUTED SENSOR MANAGEMENT ALGORITHMS

Due to the binary nature of the decision variables $\{a_{ij}(t)\}$, the sensor allocation formulation in (13) is a multidimensional sum assignment problem that is generally NP-hard [33]. This section provides a relaxed reformulation and its solution implemented in a distributed manner.

A. Convex Relaxation and KKT Conditions

To efficiently solve the problem in (13), we apply convex relaxation to the integer constraint (13d), replacing it by the linear inequality constraints $0 \leq a_{ij}(t) \leq 1$, $\forall i,j$. By doing so, (13) is relaxed to be a convex optimization problem as follows:

$$\max_{\{a_{ij}(t)\}_{i,j}} \quad \sum_{j=1}^{N_a} J_j(t) \tag{15a}$$

$$\text{s.t.} \quad \sum_{i\in\Omega_j(t)} a_{ij}(t) \geq 1, \qquad \forall j = 1,\ldots,N_a \tag{15b}$$

$$\sum_{j\in\Pi_i(t)} a_{ij}(t) \leq N_i, \qquad \forall i = 1,\ldots,N_s \tag{15c}$$

$$0 \leq a_{ij}(t) \leq 1, \qquad \forall i = 1,\ldots,N_s, \quad j = 1,\ldots,N_a. \tag{15d}$$

Note that the relaxed problem is not equivalent to the original problem (13) because the optimal value a_{ij}^* in (15) can be fractional. In fact, the optimal objective value of the relaxed problem is an upper bound of the optimal objective value of the original problem (13), because (15d) yields an enlarged feasible set encompassing that in (13d).

Next we develop a distributed algorithm to implement the optimal solution to the relaxed reformulation in (15). To this end we apply the primal-dual method along with the projection subgradient technique [34]. We first describe the Karush-Kuhn-Tucker (KKT) conditions for the primal and dual problem of (15). Let σ_j and l_i be the Lagrange multipliers corresponding to the constraints (15b) and (15c), respectively. The Lagrangian function associated with (13) can be written as

$$L(\{a_{ij}(t);l_i,\sigma_j\}_{i,j}) = \sum_{j=1}^{N_a} J_j(t) + \sum_{i=1}^{N_s} l_i \left(\sum_{j\in\Pi_i(t)} a_{ij}(t) - N_i \right) + \sum_{j=1}^{N_a} \sigma_j \left(1 - \sum_{i\in\Omega_j(t)} a_{ij}(t) \right). \quad (16)$$

Correspondingly, the Lagrange dual function is given by

$$g(\{l_i,\sigma_j\}_{i,j}) = \max_{0\le a_{ij}(t)\le 1} L(\{a_{ij}(t);l_i,\sigma_j\}_{i,j}) \quad (17)$$

for $l_i \ge 0$, $\sigma_j \ge 0$, $i = 1\ldots N_s$, $j = 1\ldots N_a$, and the optimal solutions to $\{a_{ij}(t)\}$ are

$$a_{ij}^*(t) = \arg\max_{\{a_{ij}(t)\}_{ij}} L(\{a_{ij}(t),l_i^*,\sigma_j^*\}_{i,j}), \quad \forall i \in [1,N_s], \quad j \in [1,N_a]. \quad (18)$$

Because the primal problem is a convex problem with differentiable objective and constraint functions, the Slater's condition is satisfied [34]. Hence, at the optimal values of the primal and dual variables, the convexity of our problem formulation guarantees zero duality gap.

From (16), by taking the first-order partial derivatives $\partial L/\partial a_{ij} = 0$, $\forall i,j$, we have

$$\mathrm{tr}(\mathbf{F}_j^{-1}\mathbf{P}_j(t \mid t-1)\mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)) + l_i - \sigma_j = 0 \quad (19)$$

where $\mathbf{F}_j = \mathbf{I} + \mathbf{P}_j(t \mid t-1)\sum_{i\in S_j(t)} a_{ij}(t)\mathbf{H}_{ij}^{\mathrm{T}}(t) \cdot \mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t)$, and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Overall, the KKT conditions are shown below:

$$\begin{aligned}
\mathrm{tr}(\mathbf{F}_j^{-1}\mathbf{P}_j(t \mid t-1)\mathbf{H}_{ij}^{\mathrm{T}}(t)\mathbf{M}_{ij}^{-1}(t)\mathbf{H}_{ij}(t))|_{a_{ij}^*(t)} + l_i - \sigma_j &= 0, \qquad i = 1\ldots N_s, \quad j = 1\ldots N_a \\
\sum_{j\in\Pi_i(t)} a_{ij}^*(t) &\le N_i, \qquad i = 1\ldots N_s \\
l_i^* &\ge 0, \qquad i = 1\ldots N_s \\
l_i^* \left(\sum_{j\in\Pi_i(t)} a_{ij}^*(t) - N_i \right) &= 0, \qquad i = 1\ldots N_s \\
\sum_{i\in\Omega_j(t)} a_{ij}^*(t) &\ge 1, \qquad j = 1\ldots N_a \\
\sigma_j^* &\ge 0, \qquad j = 1\ldots N_a \\
\sigma_j^* \left(1 - \sum_{i\in\Omega_j(t)} a_{ij}^*(t) \right) &= 0, \qquad j = 1\ldots N_a.
\end{aligned} \quad (20)$$

The KKT conditions provide necessary and sufficient conditions for the optimality of the primal and dual values. On the other hand, the optimal values $\{a_{ij}^*(t)\}$ cannot be obtained in a closed form from the KKT conditions. Next, we apply an iterative subgradient search algorithm to obtain the optimal solutions.

B. Distributed Algorithm Design

Let $[x]_+ = \max(x,0)$ denote the projection onto a nonnegative real value, and $[x]_0^1$ denote the projection of x onto $[0,1]$, that is, $[x]_0^1 = \{0,x,1\}$ for $x \le 0$, $0 < x < 1$ and $x \ge 1$, respectively. At each time t the decision variables and Lagrange multipliers are initialized from $k = 0$, and updated from the kth step to the $(k+1)$th one by iteratively performing projection subgradient search with a small stepsize α, as follows (the time index t is omitted during iterations without causing ambiguity):

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij} = 0, & j \notin \Pi_i(t) \\ [a_{ij}^{(k)} + \alpha(g_{ij}(\{a_{ij}^{(k)}\}_i) - l_i^{(k)} + \sigma_j^{(k)})]_0^1, & j \in \Pi_i(t) \end{cases} \quad (21a)$$

$$l_i^{(k+1)} = \left[l_i^{(k)} + \alpha \left(\sum_{j=1}^{|\Pi_i(t)|} a_{ij}^{(k)} - N_i \right) \right]_+, \qquad i \in \Omega_j(t) \tag{21b}$$

$$\sigma_j^{(k+1)} = \left[\sigma_j^{(k)} + \alpha \left(1 - \sum_{i=1}^{|\Omega_j(t)|} a_{ij}^{(k)} \right) \right]_+, \qquad j \in \Pi_i(t) \tag{21c}$$

where $g_{ij}(\{a_{ij}^{(k)}\}_i) = \mathrm{tr}(\mathbf{F}_j^{-1}(\{a_{ij}^{(k)}\}_i)\mathbf{Y}_j^{-1}(t \mid t-1)\mathbf{H}_{ij}^{\mathrm{T}}(t) \cdot \hat{\mathbf{M}}_{ij}^{-1}(t \mid t-1)\mathbf{H}_{ij}(t))$, and $\mathbf{F}_j(\{a_{ij}^{(k)}\}_i) = \mathbf{I} + \mathbf{Y}_j^{-1}(t \mid t-1) \cdot \sum_{i\in\Omega_j(t)} a_{ij}^{(k)}\mathbf{H}_{ij}(t)\mathbf{M}_{ij}^{-1}(t \mid t-1)\mathbf{H}_{ij}(t)$. After defining $\mathbf{\Phi}_{ij}(t) = \mathbf{Y}_j^{-1}(t \mid t-1)\mathbf{H}_{ij}(t)\hat{\mathbf{M}}_{-1}(t \mid t-1)\mathbf{H}_{ij}(t)$, $\forall i, j$, $g_{ij}(\cdot)$ and $\mathbf{F}_j(\cdot)$ can be rewritten as $g_{ij}(\{a_{ij}^{(k)}\}_i) = \mathrm{tr}(\mathbf{F}_j^{-1}(\{a_{ij}^{(k)}\}_i)\mathbf{\Phi}_{ij}(t))$, and $\mathbf{F}_j(\{a_{ij}^{(k)}\}_i) = \mathbf{I} + \sum_{i\in\Omega_j(t)} a_{ij}^{(k)}\mathbf{\Phi}_{ij}(t)$. Here, $\mathbf{\Phi}_{ij}(t)$ only needs to be computed once for each t, and remains unchanged during the iteration steps indexed by k.

Because the objective function in (13a) is differentiable, the above subgradient search algorithm is theoretically guaranteed to converge to the optimal values, i.e., $\lim_{k\to\infty} a_{ij}^{(k)} = a_{ij}^*(t)$, $\forall i, j$, given that the step size is small enough [34]. However, we may have limited processing time in real-time applications, and not all a_{ij} may synchronously reach their optimal values within a limited period. Thus in practical implementation, we terminate the iterations using the commonly used stopping rules: the iteration terminates at the Kth step for sensor i when the difference $\sum_{j\in\Pi_i(t)} |a_{ij}^{(K)} - a_{ij}^{(K-1)}|$ falls below a small threshold ϵ_{th} for termination, or when a maximum number of iterations $K = K_{\max}$ has been reached. At the end, any fractional final value is rounded to 0 or 1, which is taken as the allocation decision at time t, that is, $a_{ij}(t) = \mathrm{round}\{a_{ij}^{(K)}\}$, $\forall i, j$.

In terms of implementation, all the sensors that have detected a target at time t in the preceding sensing module would participate in the sensor management module, that is, $i \in \Omega(t) = \bigcup_{j=1}^{N_a} \Omega_j(t)$ ($\bigcup$ denotes union). As indicated by the distributed nature of (21), each sensor i does not need to know the total number of targets $N_a = |\bigcup_{i=1}^{N_s} \Pi_i(t)|$, as long as it knows its own sensed target subset $\Pi_i(t)$. All sensors start with an initiation phase, within which each sensor i computes the local quantity $\mathbf{\Phi}_{ij}$ for each $j \in \Pi_i(t)$, and passes on $\mathbf{\Phi}_{ij}(t)$ to its one-hop neighbors $i' \in \Omega_j(t)$, $i' \neq i$, $\forall j$. During the ensuing iterative decision updating phase, at the end of the kth iteration, each sensor $i \in \Omega(t)$ would have collected all the needed allocation decisions $\{a_{ij}^{(k)}\}_{j\in\Pi_i(t)}$ via local broadcasting, and have computed $\{\sigma_j\}_{j\in\Pi_i(t)}$ and stored them locally. At the beginning of the $(k+1)$th iteration, in a round-robin manner, each sensor $i \in \Omega(t)$ carries out the three steps in (21) to update all the decision variables and multipliers. The updated decisions $\{a_{ij}^{(k+1)}\}_j$ will then be broadcast so as to be heard by all neighboring nodes within its communication range r_C. Local broadcasting is adequate, because we have assumed that the one-hop communication range r_C is twice the sensing range r_S. As a result, any two sensors i, $i' \in \Omega_j(t)$ sensing the same target j will be less than r_C away from the target, and hence their distance is less than the one-hop range r_C. Overall, the iterative steps in (21) can be implemented in a distributed manner via one-hop local broadcasting only. Sensors do not need to know other sensors' positions and relative topology with targets. Implementation steps of the proposed distributed algorithm are summarized in Table I.

It is worth mentioning that the distributed sensor allocation algorithm can also be adopted for centralized decision making for computational convenience. In that case, a sensor i in the group $S_j(t)$ can be elected as a central controller to implement the sensor management module. All sensors pass on their messages $\mathbf{\Phi}_{ij}(t)$ to this sensor, which carries out the steps in (21) until convergence. During the iterations, there is no need for local broadcasting. Upon convergence, the central controller passes on the allocation decisions $a_{ij}(t)$ to those allocated sensors. The distributed algorithm (21) is adopted as a scalable solution to implement the centralized formulation (15), which is a multivariate optimization problem and not straightforward to solve.

C. Evaluation of Computational Complexity and Communication Costs

It is of interest to evaluate the computational complexity of the proposed distributed algorithm. In the search steps in (21), all the updating steps are scalar-based additions and projections, except for the computation of g_{ij} which requires matrix operations. The matrices $\{\mathbf{\Phi}(t)\}_{ij}$ are computed once for each t, at a flop count of $\mathcal{O}(\sum_{j=1}^{N_t}(|\Omega_j|)$. After acquiring $\{\mathbf{\Phi}_{ij}(t)\}$, g_{ij} is computed for each step k, at a flop count on the order of $\mathcal{O}(N_s N_a)$ per iteration. Meanwhile, given $\{g_{ij}\}$, the computations of a_{ij}^{k+1}, l_i^{k+1} and σ_j^{k+1} in (21) cost $\sum_{i=1}^{N_s}(4|\Pi_i|)$, $\sum_{i=1}^{N_s}(|\Pi_i| + 3)$, and $\sum_{j=1}^{N_t}(|\Omega_j| + 3)$ operations, respectively. Generally, dozens of iterations are enough based on extensive simulation cases we have tested. Thus the overall cost of the algorithm is $O(N_s N_a)$ for all N_s sensors, which is scalable for a large-size network with a large value of N_s. Table II compares the computational complexity order of the proposed algorithm against existing techniques including heuristic search [18, 19], Newton's method [25], and global search [18, 24].

Communication and broadcasting consume bandwidth and energy in a wireless sensor network. Thus, the communication load is an important concern for the distributed algorithm design. In the sensor

TABLE I
Distributed Implementation of the Proposed MTT System with Optimized Sensor Management

At time t: $\{\hat{\mathbf{u}}_j(t-1\mid t-1), \mathbf{Y}_j(t-1\mid t-1)\}_{j=1}^{N_a}$ have been obtained by the network for the groups $S_j(t)$, $\forall j$;

Sensing Module

Each sensor i detects $j \in \Pi_i(t)$ within its sensing range r_S based on the received SNR, $\forall i$. All the sensors that can detect at least one target form the sensor set $\Omega(t)$;

Sensor Allocation Module

Initialization: At $k=0$, for each sensor $i \in \Omega(t)$

- computes the target location estimate $\mathbf{p}_{a,j}$ indicated by $\hat{\mathbf{u}}_j(t\mid t-1) = \mathbf{A}_j(t)\hat{\mathbf{u}}(t-1\mid t-1)$, and the CRB matrix $\hat{\mathbf{M}}_{ij}(t\mid t-1)$ as a function of the sensor location $\mathbf{p}_{s,i}$ [29], $\forall j \in \Pi_i(t)$;
- predicts $\mathbf{Y}_j(t\mid t-1)$ from (6b), computes $\mathbf{\Phi}_{ij}(t) = \mathbf{Y}_j^{-1}(t\mid t-1)\mathbf{H}_{ij}^{\mathrm{T}}\hat{\mathbf{M}}(t\mid t-1)\mathbf{H}_{ij}(t)$, and broadcasts $\mathbf{\Phi}_{ij}(t)$ to its one-hop neighbors $i' \in \Omega_j(t)$, $\forall j \in \Pi_i(t)$;
- initializes with $a_{ij}^{(0)} = 0$, $\forall j$; $l_i^{(0)} = 0$, and $\sigma_j^{(0)} = 0, \forall j \in \Pi_i(t)$.

Iteration: Do $k \longrightarrow k+1$

For sensor $i \in \Omega(t)$,

- computes $g_{ij}(\{a_{ij}^{(k)}\}_{j\in\Pi_i(t)})$ from $\{\mathbf{\Phi}_{ij}(t)\}_j$;
- computes $\{a_{ij}^{(k+1)}\}_{j\in\Pi_i(t)}$, $l_i^{(k+1)}$ and $\{\sigma_j\}_{j\in\Pi_i(t)}$ from (21);
- broadcasts $a_{ij}^{(k+1)}$ to its one-hop neighbors $i' \in \Omega_j(t)$, $\forall j \in \Pi_i(t)$;

End For i

End k at $k=K$ when $\sum_{j\in\Pi_i(t)} |a_{ij}^{(K)} - a_{ij}^{(K-1)}| \le \epsilon_{th}$ or $K = K_{\max}$

Decision: Allocation decisions are $a_{ij}^*(t) = \mathrm{round}\{a_{ij}^{(K)}\}$, $\forall i, j$, which form the groups $S_j(t) = \{\forall i : a_{ij}^*(t) = 1\}$, $\forall j \in [1, N_a]$.

Measurement Collection Module

For each selected sensor $i \in S_j(t)$ for some j, it collects TOA and DOA measurements $(\hat{\tau}_{ij}, \hat{\theta}_{ij})$ from the allocated target j with $a_{ij}(t) = 1$, and computes the position measurements $\mathbf{z}_{ij}(t) = (\hat{x}_{ij}, \hat{y}_{ij})$ from (2);

Track Fusion Module

One elected sensor node, which acts as a local fusion center, updates the state vectors and covariance matrices $\hat{\mathbf{u}}_j(t\mid t)$ and $\mathbf{Y}_j(t\mid t)$, $\forall j$, using the sequential KF implementation of (6).

Information Relay Module

For each $j \in [1, N_a]$, a sensor node belonging to $S_j(t)$ is elected as the relaying node whose position is closest to the target position indicated by $\hat{\mathbf{u}}(t\mid t)$. The relaying node broadcasts $\hat{\mathbf{u}}_j(t\mid t)$ and $\mathbf{Y}_j(t\mid t)$. When the target does not move abruptly, the broadcast ensures relaying of necessary information to all sensors in $\Omega_j(t+1)$ of the next time, without needing to know $\Omega_j(t+1)$.

TABLE II
Complexity Comparison

Global Search	Greedy Search/ Heuristics	Newton's	Proposed Algorithm
$O(2^{N_s}N_a)$	$O(N_s^2 N_a)$	$O(N_s^3 N_a)$	$O(N_s N_a)$

allocation module, the initialization process requires each node to broadcast $\mathbf{\Phi}_{ij}(t)$ to its one-hop neighbors, which leads to the number of N_s initialization messages. The search process requires that each selected sensor $i \in \Omega(t)$ broadcast the temporary allocation variables $a_{ij}^{(k)}$ one-hop at each iteration. Assuming $K_{ij}(t)$ is the number of iterations before termination, there is a total number of $N_s + |\Omega(t)|K_{ij}(t)$ messages transmitted in the network during the allocation process at t. Apparently, the communication cost of our decentralized algorithm is linear in the network size N_s. As we show later via simulations, our decentralized algorithm typically converges in dozens of iterations, which guarantees the energy efficiency for large networks. The communication costs may be further reduced in practical tracking systems. For example, if each sensor node has knowledge about its neighbors, then sensors can compute $\Phi_{ij}(t)$ for their neighbors, without having to communicate these quantities. Further, for the single target case where $N_a = 1$, the allocation variables can be computed without any communication between sensor nodes.

In comparison, a centralized implementation requires that all selected sensors send their measurements to a fusion center via multi-hop communications. The average number of hops needed per sensor is proportional to the network size N_s, which means that the total energy consumption is on the order of $\mathcal{O}(N_s^2)$ or $\mathcal{O}(|\Omega(t)|N_s)$. Considering that the main source of energy usage resides in radio transmissions, the choice between centralized and decentralized algorithms depends on which one consumes more transmission energy, as far as the network lifetime is concerned. The energy comparison is system dependent, especially for small-scale networks. In a centralized implementation, the average energy consumption is dominated by the network size. For a decentralized implementation, the energy

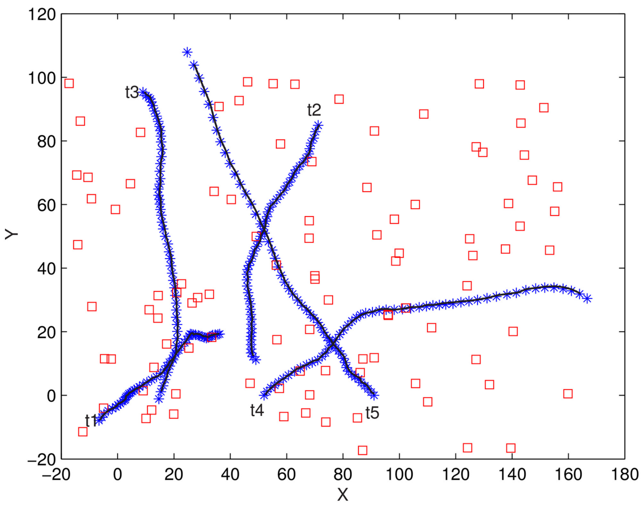

Fig. 1. Comparison of original and simulated trajectories of targets. Total number of $N_s = 100$ sensors are randomly deployed, indicated by rectangles. The 5 targets are marked as $t1$, $t2$, $t3$, $t4$, and $t5$, respectively. Dashed lines with $*$ are simulated trajectories of proposed technique, and solid lines are true target trajectories that nearly overlap with dashed lines.

consumption depends heavily on the number of iterations of searching. Therefore, we cannot claim that one is superior to the other in all scenarios. In a small network, the centralized algorithm can be more advantageous in energy saving. However, in a large network with hundreds of sensor nodes or/and a very large coverage area, the decentralized algorithm is more energy efficient because the communication and computation costs are scalable to the network size.

V. SIMULATION RESULTS

This section provides simulations to demonstrate the effectiveness of our proposed MTT system with distributed sensor management. There are N_s sensors tracking $N_a = 5$ moving targets, where N_s is chosen in the range of $[100, 500]$ in our simulations. Each sensor node can monitor a circle with a radius of 25 m. The target trajectories obey the linear dynamic model in (4), with parameters given by

$$\mathbf{A}_j(t) = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{C}_j(t) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \tag{22}$$

where the sampling interval is $\Delta t = 1$ s. The process noise is Gaussian distributed with $\mathbf{v}_j(t) \sim N(\mathbf{0}, \delta_v^2 \mathbf{I})$, with $\delta_v^2 = 0.05$.

The N_s sensors are randomly deployed in a $[-20, 160] \times [-20, 100]$ (m^2) rectangular sensing area. Each sensor node is assumed to follow the same model settings for sensing, measurement, and KF. During sensing the SNRs are collected by assuming that the received power at a sensor obeys the typical path loss model with power attenuation proportional to r^{-n}, where $r = c\tau$ is the distance between of the source and the sensor and n is the path loss exponent set to $n = 2$. During measurement collection, each sensor independently measures both the TOA and DOA parameters using a uniform linear array with $N = 4$ antenna elements and an effective signal bandwidth of $W = 10$ MHz, which are used to calculate the CRBs $\mathbf{M}_{ij}(t)$ using the formula in [29]. All sensor nodes have the maximal capacity of tracking $N_i = 2$ targets, while each target is allowed to be tracked by more than one sensors. For the subgradient search used, the step size α is 0.1, and the stopping criteria use $\epsilon_{th} = 0.0001$ and $K_{\max} = 500$. For each test case, a total of 1000 Monte Carlo trials are performed to assess the average performance of interest.

A. Tracking Performance

Figure 1 depicts the estimated trajectories of all targets using our MTT system, with reference to the true trajectories. The distributed sensor management and target tracking algorithm in Table I is tested. In the subgradient search algorithm, the Lagrangian multiplier σ_j, l_i and the allocation variables a_{ij} are initialized to be zeros for any i and j. The searching step α is set as 0.1.

To quantify the tracking performance, we investigate the cumulative distribution function (cdf) of the position errors for all targets based on Monte Carlo trials. This metric illustrates how likely an estimation error level would occur. We compare the two cases of without and with track fusion, corresponding to adopting $\sum_{j \in \Pi_i(t)} a_{ij}(t) = 1$ in (14b) and $\sum_{j \in \Pi_i(t)} a_{ij}(t) \geq 1$ in (13b), respectively. The cdf curves for all the targets are depicted in Fig. 2. In this figure the solid line indicates the cdf results of the centralized optimal allocation of the original integer programming problem (13). The dashed line represents the cdf from the distributed subgradient search algorithm for the convex relaxation problem (15) where track fusion is applied, while the dashed-dotted line with circles represents the simulated cdf without track fusion. It shows that the moving targets can be tracked accurately, and confirms that tracking with fusion achieves smaller position estimation errors with higher probability than tracking without fusion.

Another performance metric of interest is the mean square errors (MSEs) of the estimated target positions. Note that the MSE of position estimates depends on the target trajectories and sensor topology. In Fig. 3 the average MSE for the estimated target trajectories

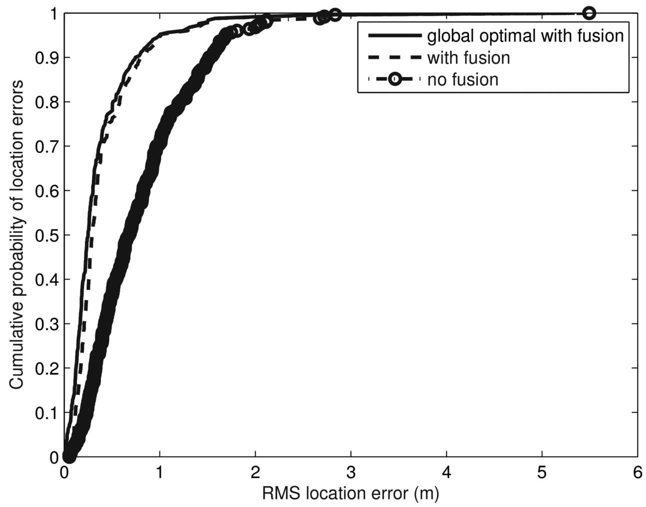

Fig. 2. Overall cdfs of position estimation errors for all targets.

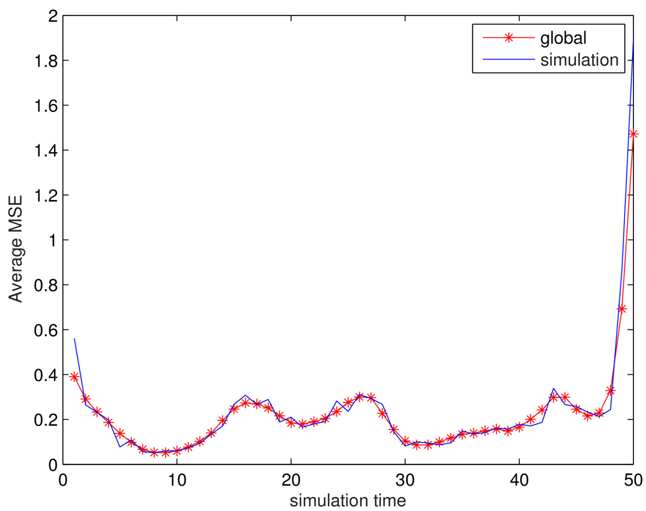

Fig. 3. Position estimation MSE averaged over for all five targets.

of all targets shown in Fig. 1 is plotted versus the tracking time. Here, each target is allowed to be tracked by multiple sensors. It illustrates the arithmetic average MSE for all targets in the system using MTT with data fusion. It is observed that the outcomes of the distributed algorithm match well with the optimal solution to the centralized original problem (13).

To illustrate the sensor allocation results, we use a small sensor network where 4 sensors are deployed at $[0, -30]$, $[100, -30]$, $[0, 70]$, and $[100, 70]$. In Fig. 4 the allocation results for the second target are depicted as an example, which describes the allocated sensor numbers over time. Meanwhile, the allocation results for sensor 2 are illustrated in Fig. 5, which describes the target numbers allocated to this sensor over time. In both cases the decentralized solutions via iterative subgradient search match well to the optimal allocation results of the original integer-valued optimization problem. It testifies that the subgradient

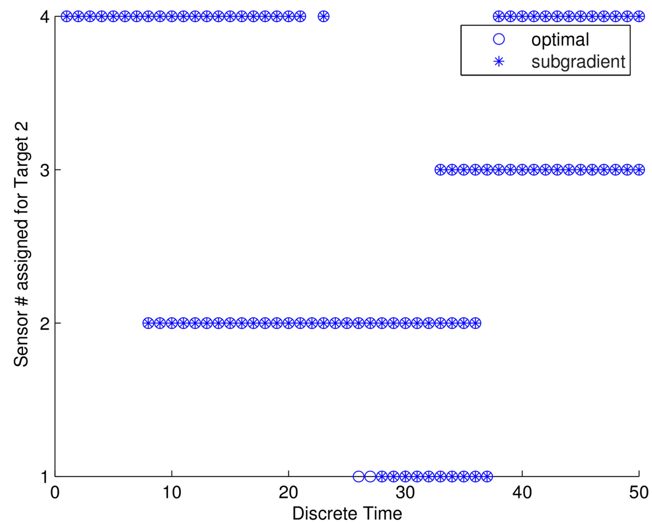

Fig. 4. Sensors assigned for target 2 in tracking. Proposed distributed subgradient method (marked by "*") nearly overlaps with optimal allocation for centralized integer-valued optimization problem (marked by "o").

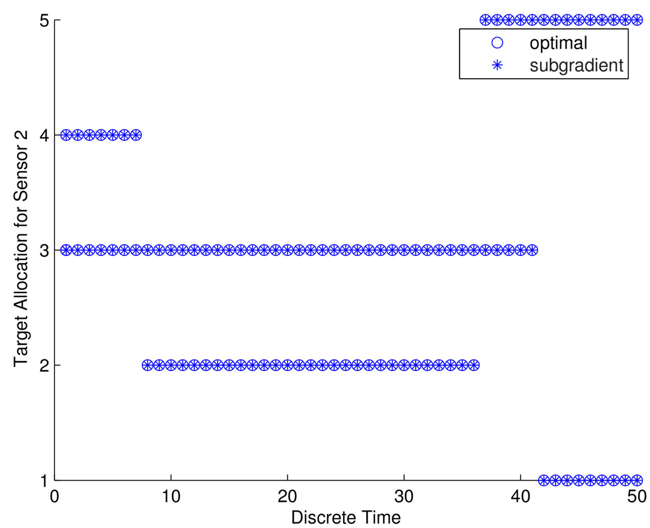

Fig. 5. Comparison of target allocation results for sensor 2 using distributed subgradient method and centralized optimal solution. Curves of two schemes are nearly identical.

search algorithm is near optimal in reaching the centralized global optimal allocation.

B. Convergence of the Distributed Algorithm

Figure 6 illustrates the convergence behavior of the subgradient search algorithm, using the same network setup as in the previous example. Interestingly, the allocation variables for the relaxed real-valued problem in (15) eventually converge to integer values 0 or 1 in most cases, given a small enough step size and adequate number of iterations. This means that there is little or no performance loss due to the convex relaxation. It is also observed that the decisions of each sensor on different targets may converge at a different pace as the targets move

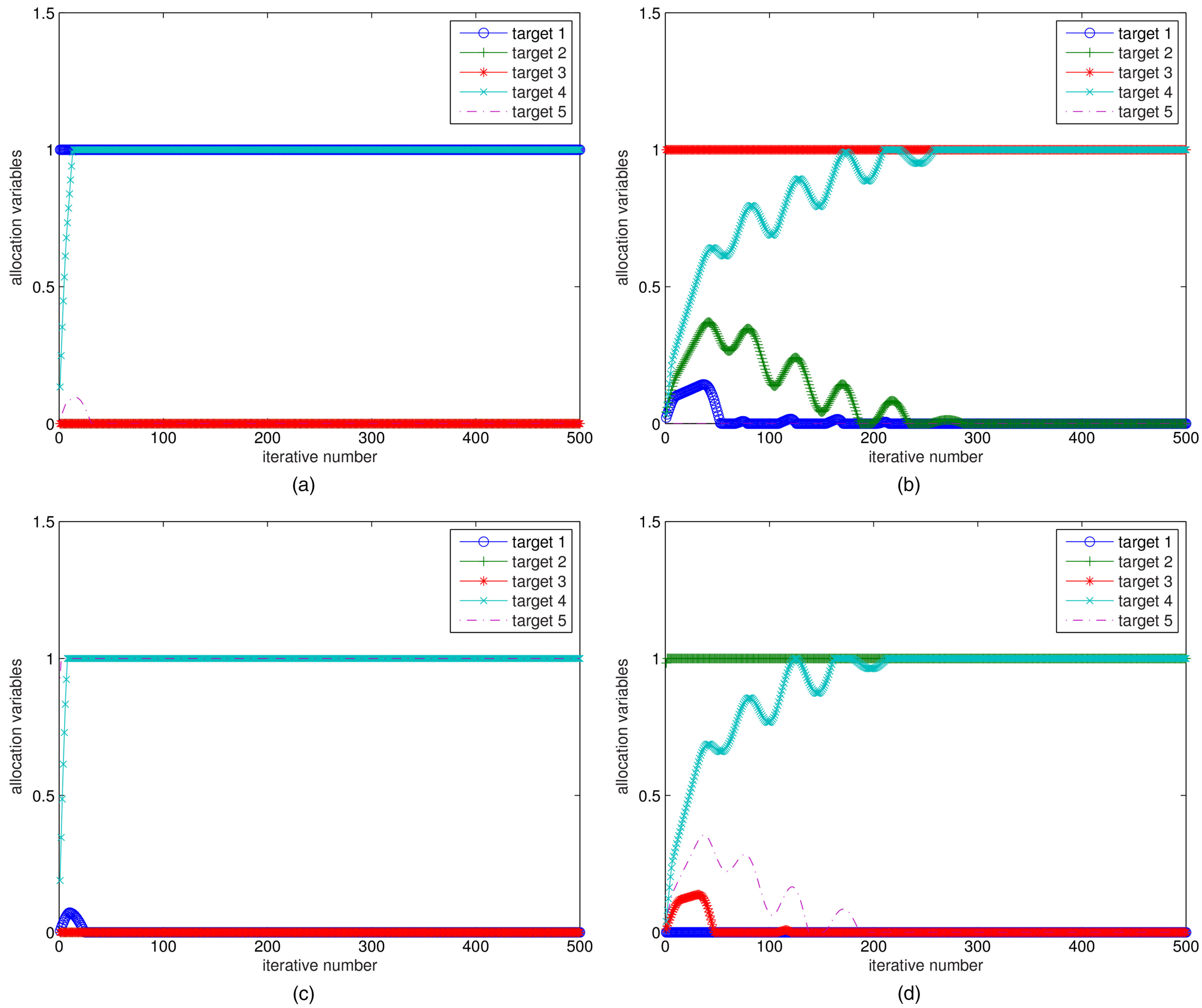

Fig. 6. Convergence behavior of all allocation variables ($\alpha = 0.1$, t=2). (a) Sensor 1. (b) Sensor 2. (c) Sensor 3. (d) Sensor 4.

over the network. Nevertheless, even well before the allocation variables converge to the binary 0 or 1, their fractional values have already converged to one side of the decision line at 0.5. Binary decisions made by the nearest neighbor rule would use 0.5 as the threshold, which apparently converges much faster to the nearest neighborhood of steady-state values, well before they converge in the MSE sense. In fact, after dozens of iterations (typically in the range of $[50, 100]$ iterations), it is usually enough for a sensor node to decide on which target to track by rounding the fractional allocation variables to binary 0 or 1. These observations assert two nice properties of the adopted convex relaxation strategy: it causes little loss of optimality compared with the original integer programming problem, and it can be coupled with the rounding operation to considerably expedite the convergence rate to the original problem. Fast convergence is essential for the overall energy efficiency of distributed implementation.

VI. SUMMARY

This paper has developed a distributed multi-sensor, MTT framework with focus on sensor-target allocation. The framework applies to a range of operating scenarios where different system configurations can be incorporated into the formulated constrained optimization problem. Distinct from previous work on sensor scheduling, the developed framework not only allows each sensor to contribute to the tracking of multiple targets, but also allows multiple sensors to track the same target and use the track fusion outcomes to dynamically guide the sensor allocation. Through convex relaxation, a linear convex problem is developed for sensor allocation, which is shown to have zero duality gap. The KKT conditions for this problem are presented, and a primal-dual solution using subgradient search is developed for distributed implementation. The proposed distributed scheme only requires local one-hop communications to save transmission power.

It is guaranteed to converge to the global optimum of the relaxed centralized allocation problem owing to its sensor cooperation, and it even approaches the global optimum of the original integer programming problem as indicated by simulations. The computational complexity and communication costs are shown to be tractable and linear in the network size, which is appealing for energy-constrained large-scale wireless networks.

There are several worthy topics for future work. Firstly, this framework can be extended to tracking systems in which sensors can only collect TOA-only and DOA-only measurements. Secondly, network energy conservation can be explicitly reflected in the sensor allocation formulations by incorporating a proper form of energy penalty on communications and computation. For example, when a penalty term is enforced on the tracking and communication costs, a sensor may choose to track less than N_i targets given the tracking constraint (13c), and the number of active sensors may be reduced. The more active sensor nodes, the better average MSE performance at the expense of higher energy consumption. While our work focuses on near-optimal sensor allocation, it also can be of interest to consider various suboptimal sensor selection methods with reduced communication costs, depending on different scenarios and energy budgets of sensing and communicating in practical tracking systems. In general, it is of practical interest to investigate the tradeoff between the average MSE in tracking and the energy requirements of various sensor management options. Thirdly, the impact of data association errors on sensor management will be assessed. In general, sensor management is relatively robust to some small or moderate level of data association errors, because the allocation decisions are integer-valued with inherent robustness to small fractional errors. Further, the impact of imperfect sensor allocation can be compensated in the ensuing tracking module in which joint data association and track fusion are performed. Hence, the impact of imperfect allocation mainly affects the energy efficiency of the network, and the impact of data association eventually has to be assessed on the tracking performance.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive feedback. Their comments have led to the simple proof for Proposition 1.

REFERENCES

[1] Liu, J., Chu, M., and Reich, J. E.
Multitarget tracking in distributed sensor networks.
IEEE Signal Processing Magazine, **24**, 3 (May 2007), 36–46.

[2] Zhan, P., Casbeer, D. W., and Swindlehurst, A. L.
A centralized control algorithm for target tracking with UAVs.
Proceedings of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, Oct. 2005, pp. 1148–1152.

[3] Oh, S., et al.
Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments.
Proceedings of the IEEE, **95**, 1 (2007), 234–254.

[4] Kreucher, C. M., et al.
An information-based approach to sensor management in large dynamic networks.
Proceedings of the IEEE, **95**, 5 (May 2007), 978–999.

[5] Lin, C-Y., Peng, W-C., and Tseng, Y-C.
Efficient in-network moving object tracking in wireless sensor networks.
IEEE Transactions on Mobile Computing, **5**, 8 (Aug. 2006), 1044–1056.

[6] Bar-Shalom, Y., Li, X-R., and Kirubarajan, T.
Estimation with Applications to Tracking and Navigation.
Hoboken, NJ: Wiley-Interscience, 2001.

[7] Kaplan, L., Bar-Shalom, Y., and Blair, W.
Assignment costs for multiple sensor track-to-track association.
IEEE Transactions on Aerospace and Electronic Systems, **44**, 2 (2008), 655–677.

[8] Martin, E. L., et al.
Distributed fusion architectures and algorithms for target tracking.
Proceedings of the IEEE, **85**, 1 (1997), 95–107.

[9] He, Y. and Chong, E. K. P.
Sensor scheduling for target tracking: A Monte Carlo sampling approach.
Elsevier Journal of Digital Signal Processing (2006), 535–545.

[10] Chhetri, A. S., Morrell, D., and Papandreou-Suppappola, A.
Sensor scheduling using a 0-1 mixed integer programming framework.
Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop, July 2006, pp. 471–476.

[11] Zhang, W. and Cao, G.
DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks.
IEEE Transactions on Wireless Communications, **11**, 5 (Sept. 2004).

[12] Zhao, F., Shin, J., and Reich, J.
Information-driven dynamic sensor collaboration.
IEEE Signal Processing Magazine, **19**, 2 (Mar. 2002), 61–72.

[13] Vercauteren, T., Guo, D., and Wang, X.
Joint multiple target tracking and classification in collaborative sensor networks.
IEEE Journal on Selected Areas in Communications, **23**, 4 (Apr. 2005), 714–723.

[14] Ng, G. W. and Ng, K. H.
Sensor management—What, why and how.
Elsevier Journal of Information Fusion, (2000), 67–75.

[15] Xiong, N. and Svensson, P.
Multi-sensor management for information fusion: Issues and approaches.
Elsevier Journal of Information Fusion, (2002), 163–186.

[16] Tharmarasa, R., Kirubarajan, T., and Hernandez, M. L.
Large-scale optimal sensor array management for multitarget tracking.
IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews, **37**, 5 (Sept. 2007), 803–814.

[17] Tharmarasa, R., Kirubarajan, T., and Hernandez, M. L.
PCRLB-based multisensor array management for multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 2 (Apr. 2007), 539–555.

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

[19] Kalandros, M., Pao, L-Y., and Ho, Y-C.
Randomization and super-heuristics in choosing sensor sets for target tracking applications.
Proceedings of the IEEE Conference on Decision and Control, vol. 2, Dec. 1999, pp. 1803–1808.

[20] Grocholsky, B., Makarenko, A., and Durrant-Whyte, H.
Information-theoretic coordinated control of multiple sensor platforms.
Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, Sept. 2003, pp. 1521–1526.

[21] Manyika, J. M. and Durrant-Whyte, H. F.
Information-theoretic approach to management in decentralized data fusion.
Proceedings of SPIE, vol. 1828, Nov. 1992, pp. 202–213.

[22] Onel, T., Ersoy, C., and Delic, H.
Information content-based sensor selection and transmission power adjustment for collaborative target tracking.
IEEE Transactions on Mobile Computing, **8** (Aug. 2009), 1103–1116.

[23] Ling, Q., Fu, Y., and Tian, Z.
Localized sensor management for multi-target tracking in wireless sensor networks.
Elsevier Journal of Information Fusion: Special Issue on Future Generation Communication Environments, (2010).

[24] Kaplan, L. M.
Global node selection for localization in a distributed sensor network.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 1 (Jan. 2006), 113–135.

[25] Joshi, S. and Boyd, S.
Sensor selection via convex optimization.
IEEE Transactions on Signal Processing, **57** (Feb. 2009), 451–462.

[26] Moses , R. L., et al.
A self-localization method for wireless sensor networks.
EURASIP Journal on Applied Signal Processing, **2003**, 4 (Mar. 2003), 348–358.

[27] Vanderveen, M. C., Papadias, C. B., and Paulraj, A.
Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array.
IEEE Communications Letters, **1**, 1 (Jan. 1997), 12–14.

[28] Picheral, J. and Spagnolini, U.
Angle and delay estimation of space-time channels for TD-CDMA systems.
IEEE Transactions on Wireless Communications, **3**, 3 (May 2004), 758–769.

[29] Fu, Y. and Tian, Z.
Cramer-Rao bounds for hybrid TOA/DOA-based location estimation in sensor networks.
IEEE Signal Processing Letters, **16**, 8 (Aug. 2009), 655–658.

[30] Gan, Q. and Harris, C. J.
Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion.
IEEE Transactions on Aerospace and Electronic Systems, **37** (Jan. 2001), 273–279.

[31] Kallianpur, G.
Stochastic Filtering Theory.
New York: Springer-Verlag, 1980.

[32] Van Trees, H. L.
Detection, Estimation and Modulation Theory, Part I.
Hoboken, NJ: Wiley-Interscience, 2001.

[33] Korte, B. and Vygen, J.
Combinatorial Optimization: Theory and Algorithms (3rd ed.).
New York: Springer, 2005.

[34] Boyd, S. and Vandenberghe, L.
Convex Optimization.
New York: Cambridge University Press, 2004.

Yinfei Fu received B.S and M.S degree from University of Electronic Science and Technology of China (UESTC) in 1997 and 2000, respectively, and Ph.D degree from Michigan Technological University in 2010.

From 2000 to 2004, he worked as a lecturer and researcher at National Key Laboratory of Science and Technology on Communications at UESTC. His research interests include statistical and adaptive signal processing, target localizaion and tracking, digital and wireless communications.

Qing Ling (M'08) received the B.S. degree in automation and the Ph.D. degree in control theory and control engineering from University of Science and Technology of China, Hefei, Anhui, in 2001 and 2006, respectively.

From 2006 to 2009, he was a postdoctoral research fellow with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton. Since 2009 he has been an associate professor with the Department of Automation, University of Science and Technology of China. His current research focuses on decentralized optimization of networked multi-agent systems.

Zhi Tian (M'98—SM'06) received the B.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1994, and the M.S. and Ph.D. degrees from George Mason University, Fairfax, VA, in 1998 and 2000.

Since August 2000, she has been on the faculty of Michigan Technological University, where she is currently a professor. Her general interests are in the areas of signal processing for wireless communications, estimation, and detection theory. Current research focuses on cognitive radio networks and distributed wireless sensor networks.

Dr. Tian served as associate editor for *IEEE Transactions on Wireless Communications and IEEE Transactions on Signal Processing*. She received a CAREER award in 2003 from the U.S. National Science Foundation.