SAND2019-11381C

# Autonomous Multi-Platform Sensor Scheduling for Intelligence, Surveillance, and Reconnaissance

John A. Richards,* Anirudh Patel, Adam Thorpe, and Rachel Schlossman
Pathfinder Technologies
Sandia National Laboratories
Albuquerque, NM

Presented at 2019 National Symposium on Sensor and Data Fusion
San Diego, CA
October 22, 2019

---

## Abstract

Efficient, effective use of sensors has long been a challenge for intelligence, surveillance, and reconnaissance (ISR) missions. While important even for traditional ISR missions, this challenge is foundational to autonomous ISR missions, in which the sensor tasking and scheduling functions traditionally performed by a human operator or analyst must be automated. To date, most consideration of sensor scheduling automation has been in the context of single-platform systems. However, the challenge of sensor-scheduling automation is even more marked, and its solution even more critical, for autonomous multi-platform systems. Effective ISR in autonomous multi-platform systems requires careful spatial and temporal coordination of sensor use across platforms, usually in the presence of stringent communication limitations. We present a novel technique enabling effective sensor tasking and scheduling for autonomous multi-platform ISR systems. This technique is applicable in a wide variety of operational contexts and for a diverse set of ISR missions. It formalizes multi-platform sensor scheduling as optimization of a utility function derived from a mission-specific measure of performance, whose calculation is feasible using a combination of sensor-specific state and ISR state shared between platforms. We employ auction methods and mixed-integer linear programming (MILP) to provide rigorous optimization of sensor use to realize maximum mission benefit, even in a dynamic battlespace and fluid operational environment characterized by the arrival and departure of sensing platforms. Our approach does not require the communication of high-bandwidth data streams or any other detailed platform or sensor information, but only compact, low-bandwidth information related to sensing utility, task assignments, and ISR state updates. Our approach is scalable with the number of platforms and sensors supporting the mission and with the number of targets or extent of the region being surveilled. We provide details of the design and implementation of our autonomous multi-platform sensor scheduling approach, including its application to specific ISR missions of interest, and show its implementation in a modeling, simulation, and visualization environment.

---

* Email: jaricha@sandia.gov

This research was supported by the Laboratory Directed Research and Development Program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. The views expressed in this article do not necessarily reflect the views of the U.S. Department of Energy or the United States Government.

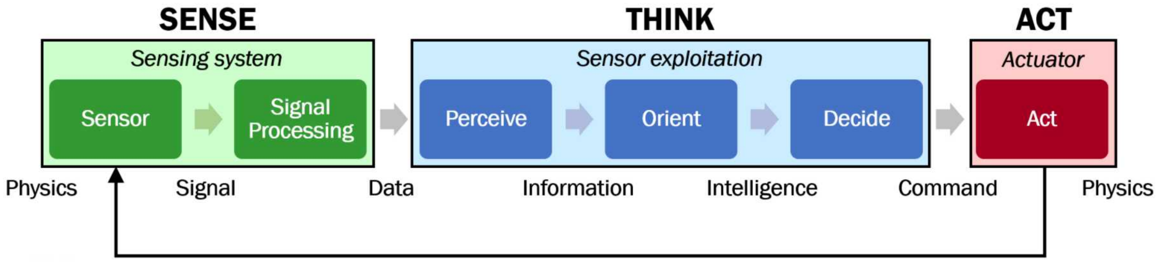


Figure 1: Sense-think-act paradigm for autonomous systems.

# 1 Introduction

Effective prosecution of intelligence, surveillance, and reconnaissance (ISR) missions requires judicious and productive use of the sensors available to support the mission. Increasing deployment of sensors and sensing platforms has greatly expanded ISR opportunities and applications, but the demand for these sensors and platforms has largely kept pace with the supply. As a result, the careful planning of sensor use is as critical now as it has ever been. Robust, reliable sensor tasking and scheduling is thus a key—and increasing—need across the spectrum of ISR applications.

While important even for traditional human-directed ISR applications, the need for effective sensor tasking and scheduling is absolutely foundational to ISR missions performed by autonomous systems. Autonomous systems seek to achieve some high-level goal without continuous human direction, replacing the traditional human *in* the loop with a human *on* the loop. As depicted in Figure 1, autonomous systems are characterized by a sense-think-act paradigm: sensors are used to transduce some physical phenomenon into a data stream ("sense") that is subjected to exploitation to extract relevant information ("think"), which provides the basis for a decision to actuate some physical change ("act"). The process depicted in Figure 1 repeats in a closed-loop fashion to drive the system toward a desired end state, with sensors and their products critical to every step. In the context of autonomous ISR, judicious sensor use is essential to ensure that sensor data is obtained on all relevant targets or regions, and that the collected data provides timely information that evolves with the engagement to support mission decisions that will yield a desired outcome. Because sensors are scarce and the data they provide is critical, sensor tasking has almost always been a human-directed mission function. However, in an autonomous ISR system, all sensor tasking and scheduling must be automated. Effective optimization of sensor tasking and scheduling is a key challenge that dictates the success or failure of autonomous ISR systems and missions.

Effective and efficient sensor use is key for any autonomous ISR system. However, it is especially critical in autonomous multi-platform systems, in which distinct sensing platforms collaborate autonomously in support of some ISR mission. Successful prosecution of ISR missions by such systems requires not just the careful management of single-point sensor resources that is required for single-platform, single-sensor missions, but also the careful spatial and temporal coordination of sensor usage across a distributed system to enable collaborative ISR. This must be done in the presence of often stringent communication limitations, in which any platform or sensor will generally have only limited knowledge of the state and capabilities of any other platform or sensor, and potentially no access to the data streams collected by other platforms and sensors. It must be done in a manner that accommodates potentially diverse capabilities and operational requirements among the collection of platforms. Unfortunately, existing and emerging sensor tasking and scheduling techniques—even those designed for use on autonomous single-platform systems—are largely inadequate for use by autonomous multi-platform systems.

We have developed an approach enabling effective, efficient use of sensors across a distributed collection of platforms to achieve autonomous multi-platform ISR. Our approach does not depend on the exchange of high-bandwidth sensor data streams or detailed platform or sensor state information or capabilities, but only on the exchange of compact and low-bandwidth information related to sensing-task mission benefit and ISR state information. It is thus applicable even in highly communication-constrained environments. Our approach supports systems comprised of either homogeneous or heterogeneous collections of platforms and sensors, and it is applicable to a wide variety of specific ISR missions. It directs sensor usage adaptively to optimize of mission benefit, even in a dynamic battlespace and in a fluid operational environment in which individual platforms and sensors may be joining or departing the collaborative mission. Our approach can be used to obtain both immediate sensor tasking assignments and longer-term sensor schedules that involve a sequence of tasks stretching into the future.

The rest of this paper is organized as follows. Section 2 describes the underlying formulation, key concepts, and the basic structure of our approach that is applicable to a wide variety of ISR missions. Section 3 provides details of the approach related to the quantification of mission benefit and the definition of relevant state for specific ISR missions of interest. Section 4 describes the optimization of mission benefit given the mission-benefit quantification of the previous section. Section 5 presents an implementation of the our approach in a modeling, simulation, and visualization environment.

## 2 Formulation

Consider a multi-platform collaborative ISR mission involving $M$ sensors, each of which may concurrently perform one of $N$ discrete sensing tasks related to the mission goal.[1] (In general, $M$ and $N$ may both vary over the course of the mission.) We seek to assign sensors to tasks over a discrete timeline consisting of $L$ successive intervals. We define a set $\mathbf{S}$ of $LMN$ task-assignment indicator variables $s_{l,m,n}$, each taking value 1 if sensor $m$ is assigned to perform task $n$ during timestep $l$, and value 0 otherwise. We allow that the selection of $\mathbf{S}$ may be constrained by a number of mission-, platform-, sensor-, and task-specific factors. Any choice of $\mathbf{S}$—that is, of the full set of $LMN$ task-assignment indicator variables $s_{l,m,n}$—that meets these constraints represents a feasible sensor schedule for the multi-platform system at hand.

We formalize multi-platform sensor scheduling as an optimization problem, in which a feasible $\mathbf{S}$ is to be selected to maximize some mission-specific measure of performance $P$ that provides a numerical representation of mission efficacy. In general, $P$ is highly mission-dependent: it depends on the specific goal of the ISR mission at hand.[2] Regardless of the details of the mission at hand, however, $P$ measures how effectively we have measured what we will term the ISR state, or the properties of the external world that are important to our stated ISR goal. We assume that at any point in the mission, $P$ may be calculated from our representation of the ISR state and its associated uncertainty. Given the current ISR state and its associated uncertainty, the future evolution of $P$ is stochastic: it will depend on the evolution of the ISR state, on the evolution of the sensing-system state, and on the choices we make about sensor tasking (that is, on $\mathbf{S}$). Our objective function for the sensor-scheduling optimization problem is the expected future value of $P$ given a sensor schedule $\mathbf{S}$ and given the current ISR state $\mathbf{X}_{\text{ISR}}$ and sensing-system state $\mathbf{X}_{\text{ss}}$, or

$$E(P|\mathbf{S}, \mathbf{X}_{\text{ISR}}, \mathbf{X}_{\text{ss}}). \tag{1}$$

[1] For missions involving ISR of distinct targets, $N$ naturally corresponds to the number of targets. For missions involving ISR of a continuous region, $N$ might correspond to a number of discretized subregions.

[2] We will investigate specific measures of performance $P$ for different ISR missions of interest in Section 3.

(Note that while the current value of $P$ will generally only depend on $\mathbf{X}_{\text{ISR}}$ and not $\mathbf{X}_{\text{ss}}$, the expectation of its future value will generally depend on both terms.) The expectation of (1) might be taken for any future time, but we will typically be interested in its value at the conclusion of all sensing tasks indicated by schedule $\mathbf{S}$.

Obviously, maximization of the objective function of (1) depends strongly on the definition of performance $P$. This is a measure of performance that will typically be specific to a particular mission. However, regardless of the choice of $P$, we will leverage the concept of *utility* to characterize its dependence on sensor-task assignments and to facilitate the optimization. We define utility to be the expected mission benefit of allocating a particular sensor to a particular task at a particular time, defined in terms of the mission-specific measure of performance. Let $\mathbf{S}_{\delta_{l,m,n}}$ be a candidate schedule consisting solely of a single sensor-task assignment: the allocation of sensor $m$ to task $n$ at timestep $l$. (Thus, each of the $LMN$ task-assignment indicator variables in $\mathbf{S}_{\delta_{l,m,n}}$ has value 0 except for the single variable corresponding to sensor $m$, task $n$, and timestep $l$, which has value 1.) Let $\mathbf{S}_0$ be a "null schedule" in which no sensors are assigned to any tasks at any time—that is, in which all $LMN$ task-assignment indicator variables have value 0. Then the utility of allocating sensor $m$ to task $n$ at timestep $l$ is taken to be[3]

$$u_{l,m,n} = E(P|\mathbf{S}_{\delta_{l,m,n}}, \mathbf{X}_{\text{ISR}}, \mathbf{X}_{\text{ss}}) - E(P|\mathbf{S}_0, \mathbf{X}_{\text{ISR}}, \mathbf{X}_{\text{ss}}). \tag{2}$$

Thus, the utility of the single sensing task indicated by $\mathbf{S}_{\delta_{l,m,n}}$ is simply the expected marginal benefit to mission performance if the given sensor performs the given task at the given time. Because we assume that any sensing tasks is performed independently by an individual sensor, the expected mission benefit of a task assigned to a particular sensor does not depend on the state of any other sensor. Denoting $\mathbf{X}_m$ to indicate the portion of $\mathbf{X}_{\text{ss}}$ that pertains to the state of sensor $m$, this means that

$$u_{l,m,n} = E(P|\mathbf{S}_{\delta_{l,m,n}}, \mathbf{X}_{\text{ISR}}, \mathbf{X}_m) - E(P|\mathbf{S}_0, \mathbf{X}_{\text{ISR}}). \tag{3}$$

Given this representation, optimization of mission benefit follows directly: we seek a schedule $\mathbf{S}$ for which the total utility, calculated as the sum of individual task-assignment utilities as in (3), is maximized.

Note that in the general formulation constructed thus far, the precise mathematical forms of $P$ and $u$, as well as the precise components of $\mathbf{X}_{\text{ISR}}$ and $\mathbf{X}_s$, remain unspecified. These will typically depend strongly on the mission and system at hand. However, the construction of (3) immediately spurs two important observations. First, given appropriate state information and a mathematical specification of how $P$ depends on that state, any sensor can, in theory, apply (3) to calculate the benefit of each of the $LN$ tasks it might perform. Second, any sensor can calculate these values solely from the ISR state and its own sensor state, without regard to the state of any other sensor. This means that no sensor-specific information need be transmitted between platforms, and leads to a natural conception of all state information being partitioned into a public component ($\mathbf{X}_{\text{ISR}}$) and $M$ distinct sensor-specific private components ($\mathbf{X}_m$).

The encapsulation of mission benefit in utility and the partitioning of state into private and public components have important implications that directly suggest our approach to multi-platform sensor scheduling. In particular, they imply that the computation required to optimize mission benefit over sensor schedule can happen in a largely distributed fashion, and with very limited information exchange between platforms. Public state must be communicated between platforms to enable the utility calculation of (3), but no private state must be exchanged. Utilities must be communicated between platforms to enable the maximization of total mission benefit across sensor schedules, but no other information must be exchanged. Additionally, assuming that each platform has sufficient computational resources to exploit its own collected data streams,

[3]Note that $P$ has been defined such that a larger value indicates better performance. If the opposite polarity holds, then the derivation the follows can be changed in obvious, trivial ways—namely, by changing signs or changing the optimization objective from maximization to minimization.

no sensor data must be transmitted off-board any platform—instead, only compact sensor-exploitation information needed to update public state must be communicated. Multi-platform sensor scheduling is thus possible in a largely distributed fashion with limited communication between platforms, and simply requires a framework to broker the required information exchanges.

Fortunately, there is an existing family of straightforward techniques that provide the framework needed here: auction methods. Auction methods involve the structured exchange of information between multiple agents through bidding to arrive at some globally most-desirable result [8]. As their name implies, auction methods are derived from the well-known practice of property sales to competing agents through monetary bids. However, auction methods have been successfully applied to problems in a number of domains that, at first glance, have little obvious connection to property transfer—and in which, like ours, the agents are not competitive but collaborative. One such example is multi-robot task allocation [6], in which a team of robots collaborate in order to achieve some desired goal. This problem is similar to ours, in that multiple agents must perform discrete tasks in order to achieve some more-encompassing result, and in which the quality of the overall result depends on the underlying assignment of tasks to agents.

Figure 2 provides a step-by-step description of our application of auction methods to multi-platform sensor scheduling. It is immediately seen to be relatively straightforward, with each agent responsible for a well-defined set of tasks, and with the flow of information between agents clearly delineated. One agent takes the role of the auctioneer, and all other agents take the role of bidders. (The auctioneer may also be a bidder.) The auctioneer is responsible for the maintenance of all public state information and for optimizing over reported utilities to determine a sensing schedule. The bidders are responsible for maintaining their own private state, and for using this private state in conjunction with the auctioneer-provided public state to calculate sensing-task utilities. Auctions are performed repeatedly, each yielding a schedule specifying the assignment of tasks to sensors over the next $L$ timesteps. If operational requirements dictate, this schedule may be implemented as-is, tying each sensor to a fixed sequence of tasks over the next $L$ timesteps. Alternatively, if long-horizon advance scheduling is not required, then each sensor may be assigned only the single task indicated in the current timestep, and then another $L$-interval auction may be performed in the next timestep. Note that in this latter case, repeated auctions provide for continual re-optimization as the ISR and sensor state evolves. Furthermore, optimization over $L$ timesteps is desirable even if sensors are to be tasked only for a single timestep, simply because it avoids potentially short-sighted, greedy taskings that will prove suboptimal as an engagement evolves.[4] Figure 3 provides an information and processing flow diagram for the auction approach, structured to indicate how each step in the process conforms to the sense-think-act paradigm for autonomous systems depicted in Figure 1.

Note that while Figure 2 provides a high-level step-by-step description of our auction methodology, it lacks many details that are essential to the actual implementation of this process for any particular mission. For instance, Step 3 is silent regarding the actual definition of utility for a particular mission (as well as the private and public state required to calculate utility), and Step 5 does not provide any details on how utility optimization may actually be performed to yield an optimal set of task assignments. Furthermore, the definition of public and private state used throughout Figure 2 is also as-yet unspecified. We examine the topics of utility and state in Section 3, and the topic of optimization in Section 4.

[4]This is analogous to model predictive control [4], in which an optimal long-term control policy is obtained and partially implemented, but in which periodic re-optimization yields new long-term control policies that supplant each previous policy.

0. **Role assignment.** One sensing platform is designated as the auctioneer; the others are designated as bidders. (The auctioneer may also serve as a bidder.)
1. **Public state dissemination.** The auctioneer transmits the latest public state information to all bidders.
2. **Auction call.** The auctioneer announces an auction, requesting that each bidder provide their utility for each of the $N$ sensing tasks at each of the $L$ schedule intervals.
3. **Utility calculation.** Each bidder uses the public state (received from the auctioneer in Step 1) and its own private state to calculate the feasibility and utility of performing each of the $N$ potential sensing tasks at each of the $L$ schedule intervals.
4. **Bid transmission.** Each bidder transmits its calculated utilities to the auctioneer as a set of $LN$ bids. (If any of the slated tasks is deemed infeasible by any bidder, it simply withholds a bid on that task.)
5. **Optimization.** The auctioneer processes the full set of $LMN$ bids to determine a set of task assignments to provide the maximum utility over the scheduling timeline.
6. **Assignment notification.** The auctioneer notifies the winning bidders of their sensing task assignments over the scheduling timeline.
7. **Task execution.** The winning bidders execute their assigned sensing tasks for the current timestep. Each bidder processes its data stream to yield information relevant to the public state.
8. **Information transmission.** Each bidder transmits the information extracted from its current-timestep sensing task in Step 7 to the auctioneer.
9. **Public state update.** The auctioneer incorporates all information from all bidders to update the public state.
10. **Time progression.** The timestep increments, and the process returns to Step 1.

Figure 2: Auction approach to multi-platform sensor scheduling.

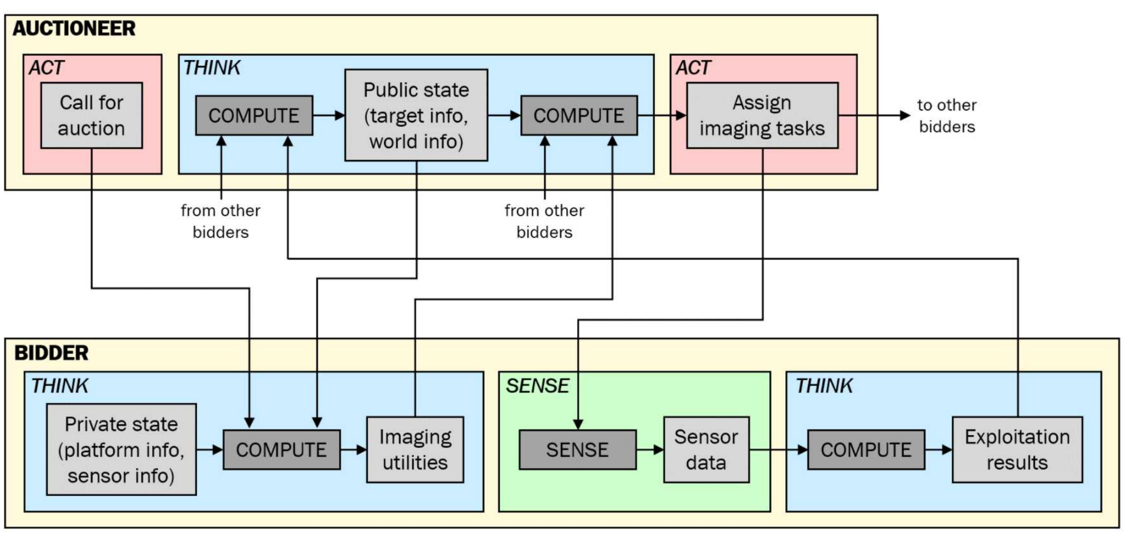


Figure 3: Information and processing flow diagram for auction approach.

# 3 Utility and State

Utility was defined in Section 2 as the expected marginal mission benefit of performing a particular sensing task by a particular sensor at a particular time, in the absence of any other task assignments. This led to the mathematical representation of (3), in which utility depends on a measure of performance $P$, given the candidate task, a public ISR state $\mathbf{X}_{\mathrm{ISR}}$, and a private sensor state $\mathbf{X}_m$. The mission at hand dictates the particular form of the measure of performance, the public and private states, and utility. This provides a flexibility that is one of the key strengths of the auction approach. It is suitable for application to a variety of missions.

In this section we examine the application of utility and public and private state information to two particular ISR missions of interest: multi-target tracking and multi-target identification. We present a manifestly reasonable measure of performance for each, and show how a precise definition of utility and a natural definition of public and private state flows from the presented performance measure. We examine the multi-target tracking mission in Section 3.1 and the multi-target identification mission in Section 3.2. In each case, we assume for convenience that the initial presence of any target of interest is provided as an external cue to our sensing system, so that our mission involves only sensing individual known targets, and not performing any wide-area search functions.[5] Modification of what follows to incorporate wide-area search tasks into schedule selection would be relatively straightforward.

## 3.1 Mission 1: Multi-Target Tracking

Many ISR missions involve tracking moving objects. Tracking typically involves repeated re-estimation of the locations and velocities of targets of interest as new sensor measurements of those targets are provided. Tracker estimates are imperfect: they depend on not only the stochastic nature of the targets themselves, but also on the characteristics of the sensors and measurements that are used to maintain the tracks. The uncertainty inherent in tracker location estimates is typically quantified in terms of an error covariance. If the tracker provides an estimate $\hat{\mathbf{x}}_n$ of the location $\mathbf{x}_n$ of the $n$th target it is tracking, then the error covariance is defined as

$$\mathbf{\Lambda}_n = E\left((\hat{\mathbf{x}}_n - \mathbf{x}_n)(\hat{\mathbf{x}}_n - \mathbf{x}_n)'\right). \tag{4}$$

The error covariance is often envisioned as an "error ellipse," or an ellipsoidal region centered on $\hat{\mathbf{x}}_n$ whose extent and orientation are determined by the eigenvalues and eigenvectors of $\mathbf{\Lambda}_n$. Because the goal of a tracking mission is typically to determine the location of objects of interest with the best possible accuracy, a natural measure of performance is provided by the total area of the error ellipses of all tracks. In terms of $\mathbf{\Lambda}_n$, this measure of performance is simply

$$P = -\sum_{n=1}^{N} |\mathbf{\Lambda}_n|^{1/2}, \tag{5}$$

where $|\bullet|$ indicates the determinant operator. Note that the leading negative sign in (5) ensures that larger $P$ indicates better performance, as indicated in Section 2.

Given the measure of performance in (5), we can apply (3) to specify the utility of any candidate sensing task. That is, we can calculate utility by computing the expected change in $P$ imparted by assigning a sensor to a target at a particular time. In general, assignment of a sensor to a particular target $n$ will, with some

[5] In the context of the multi-target tracking mission, wide-area search is needed for track initiation. In the context of multi-target identification, wide-area search is needed for target detection.

probability, yield a location measurement that can be incorporated into the target track to reduce the single covariance matrix $\mathbf{\Lambda}_n$.[6] The extraction of a target measurement and the estimation of target location from an extracted measurement are both stochastic processes, and they depend on various sensor and external characteristics. Let us denote the probability of successful extraction of a target measurement (that is, the probability of detection) by $p_\mathrm{d}$, and let the covariance of the target location estimated from the extracted measurement by $\mathbf{R}$.[7] The incorporation of a new location measurement into a target track is highly tracker-dependent. In practice, many trackers are Kalman filters [5]. Kalman-filter trackers incorporate new location measurements into existing tracks according to a weighting determined from $\mathbf{\Lambda}_n$ and $\mathbf{R}$ to provide an updated track location estimate with reduced covariance of

$$\mathbf{\Lambda}_n - \mathbf{\Lambda}_n \left(\mathbf{R} + \mathbf{\Lambda}_n\right)^{-1} \mathbf{\Lambda}_n. \tag{6}$$

Application of (5) and (6) to (3) can be shown [13] to give a utility definition that is simply

$$u = p_\mathrm{d} \frac{|\mathbf{\Lambda}_n|}{|\mathbf{\Lambda}_n + \mathbf{R}|^{1/2}}. \tag{7}$$

Here, the ratio of determinants captures the reduction in the area of the $n$th error ellipse if the sensing task yields a new measurement of the location of target $n$, and the leading $p_\mathrm{d}$ term captures the probability that a new measurement will be successfully extracted if the sensing task is performed. Both of these terms follow from the probabilistic nature of the exploitation of the data produced by the assigned sensing task. As a whole, (7) provides an unambiguous, compact, and easily computable utility for any candidate sensing task in a mult-target tracking mission.

The utility definition of (7) depends on three quantities: a track error covariance $\mathbf{\Lambda}_n$, a sensor measurement covariance $\mathbf{R}$, and a sensor probability of detection $p_\mathrm{d}$. The $\mathbf{\Lambda}_n$ term is independent of the sensor, and in fact, must be available to all sensors to compute their utilities. This term thus clearly belongs to the public state $\mathbf{X}_\mathrm{ISR}$, along with the actual track location estimates $\hat{\mathbf{x}}_n$ themselves. In contrast, $\mathbf{R}$ and $p_\mathrm{d}$ each depend strongly on characteristics and capabilities of the sensor at hand. For instance, $\mathbf{R}$ and $p_\mathrm{d}$ will usually depend on the sensor location and orientation with respect to the target, as well as on various other factors. Whatever intrinsic factors an individual sensor $m$ requires to represent its own $\mathbf{R}$ and $p_\mathrm{d}$ for a candidate sensing task thus belong to its individual private state $\mathbf{X}_m$. All of this intrinsic information resides solely on the sensor itself, and need never be communicated to any other sensor. As per the framework of Figure 2, each sensor need only ever transmit the utilities it calculates as in (7) and the target-location measurements it extracts from the data it collects. In return, each sensor need only receive target location and covariance updates and sensing task assignments.

### 3.2 Mission 2: Multi-Target Identification

Many ISR missions use sensors to perform automatic target recognition (ATR)—that is, to identify targets by type or class. Because sensing and exploitation are inherently imperfect and stochastic, unambiguous target identification is not possible. Effective ATR thus involves not merely declaration of target type or class, but also assessment of the probability of target identity from among a set of hypotheses. Target identification

[6]Closely spaced targets or wide-field-of-view sensors may result in a single task assignment producing location measurements for multiple targets that, in turn, affect multiple covariance matrices. If needed, this possibility can be built into our formulation, but for the sake of simplicity it is not done here.

[7]In what follows, we assume that the coordinate frame used for extracted-measurement location estimates is the same as the coordinate frame used for track locations. Accommodation of different coordinate frames is straightforward but complicates the subsequent exposition by requiring introduction of additional terms needed for frame conversions.

is often posed as the assessment of target identity from among a set of $K$ mutually exclusive, collectively exhaustive hypotheses.[8] In this context, the goal of an ATR system is to report, for each target of interest $n$, a vector of proabababilities

$$\mathbf{p}_n = [p_{n,1}, \ldots, p_{n,K}], \tag{8}$$

where each $p_{n,k}$ represents the probability that target $n$ has an identity of type $k$, with

$$0 \leq p_{n,k} \leq 1 \tag{9}$$

and

$$\sum_{k=1}^{K} p_{n,k} = 1. \tag{10}$$

In general, $\mathbf{p}_n$ evolves through the course of an engagement, as additional measurements of the target are obtained by one or more sensors. Each successive sensor measurement $i$ of target $n$ is subjected to exploitation to yield a single-look probabilisitic assessment $\boldsymbol{\pi}_n^{(i)}$. This must be fused with the previous probabilistic assessment $\mathbf{p}_n^{(i-1)}$ that incorporates information from the first $i-1$ sensor measurements to yield an updated probabilistic assessment $\mathbf{p}_n^{(i)}$ that incorporates information from all $i$ measurements. This is usually performed according to an iterative update rule

$$\mathbf{p}_n^{(i)} = f(\mathbf{p}_n^{(i-1)}, \boldsymbol{\pi}_n^{(i)}), \tag{11}$$

where the iteration is initialized with a prior $\boldsymbol{\pi}_n^{(0)}$. If successive sensor looks provide new information, if the ATR algorithms are effective in extracting that information, and if the update rule $f(\bullet)$ in (11) is robust and realistic, then $\mathbf{p}_n^{(i)}$ should evolve to reflect progressively more certainty about the identity of the target as an engagement proceeds.

Given that the goal of a multi-target identification mission is to determine the identities of targets with the greatest possible certainty, a natural measure of performance for such a mission is provided by entropy [3]. The entropy of a discrete probability distribution is defined as

$$H(\mathbf{p}) = -\sum_{k=1}^{K} p_k \log_2 p_k, \tag{12}$$

where $p_k \log_2 p_k$ is taken to be zero for zero-valued $p_k$. Entropy as defined in (12) is measured in bits, and it indicates the uncertainty inherent in a sample drawn from the distribution in question. A reduction in entropy corresponds to a reduction in uncertainty. The entropy of a $K$-element probabilistic assessment $\mathbf{p}$ is bounded below by zero (achieved when one $p_k$ is unity-valued and the rest are zero-valued) and is bounded above by $\log_2 K$ (achieved when $\mathbf{p}$ is uniform). We will take the measure of performance for a multi-target identification mission to be

$$P = -\sum_{n=1}^{N} H(\mathbf{p}_n), \tag{13}$$

or the negative of the total entropy in the probabilistic assessments of all targets. Note that the leading negative sign in (13) ensures that a greater $P$ indicates better performance, as assumed in Section 2.

[8] Typically, $K-1$ of these hypotheses correspond to specific target types or classes of interest (thus satisfying the mutual-exclusion requirement) and the $K$th hypothesis corresponds to "none of the above" (thus satisfying the collective-exhaustion requirement).

Given the measure of performance of (13), the utility of any sensing task is given by application of (3). In general, assignment of a sensor to a single target will have the potential to impact only that target's probabilistic assessment[9], subject to the successful extraction of the target from the collected data. The impact on entropy will depend on the quality of the probabilistic assessment and on the form of the update rule $f(\bullet)$ of (11). Given a straighforward model [11] for the quality and distribution of single-look probabilistic assessments that is characterized by a probability of correct identification $p_{\mathrm{id}}$ and a set of $K$ representative single-look probabilistic assessments $\bar{\boldsymbol{\pi}}_k$ corresponding to each identity hypothesis, this utility can be shown [12] to be

$$u = \sum_{k=1}^{K} \left[ p_{\mathrm{id}} p_{n,k} + \frac{1-p_{\mathrm{id}}}{K-1}(1-p_{n,k}) \right] \left( H(\mathbf{p}_n) - H(f(\mathbf{p}_n, \bar{\boldsymbol{\pi}}_k)) \right). \tag{14}$$

The utility of (14) is simply the expected reduction in entropy if sensor $m$ is tasked to obtain another measurement of target $n$.

While not quite as spare or elegant as the multi-target tracking utility of (7), the multi-target identification utility of (14) is also a straightforward function of a relatively small number of terms that depend on the ISR state and the sensor state: namely, the probalistic assessment $\mathbf{p}_n$, a set of nominal sensor-specific single-measurement probabilistic assessments $\bar{\boldsymbol{\pi}}_k$, and a sensor-specific probability of correct identification $p_{\mathrm{id}}$. The $\mathbf{p}_n$ term is independent of the sensor, and in fact, must be available to all sensors to compute their utilities. This term thus clearly belongs to the public state $\mathbf{X}_{\mathrm{ISR}}$. In contrast, $\bar{\boldsymbol{\pi}}_k$ and $p_{\mathrm{id}}$ each depend strongly on characteristics and capabilities of the sensor at hand. Whatever intrinsic factors an individual sensor $m$ requires to model its own $\bar{\boldsymbol{\pi}}_k$ and $p_{\mathrm{id}}$ for a candidate sensing task thus belong to its individual private state $\mathbf{X}_m$. As before, all of this intrinsic, sensor-specific information resides solely on the sensor itself, and need never be communicated to any other sensor. As per the framework of Figure 2, each sensor need only ever transmit the utilities calculated according to (14) and the probabilistic assessments $\boldsymbol{\pi}_n$ it extracts from the data it collects. In return, each sensor need only ever receive updates of the overall probabilistic assessments $\boldsymbol{\pi}_n$ and sensing task assignments.

### 3.3 Other Missions

The previous two sections detail the derivation of utility functions and the definition of public and private state for two particular ISR missions of interest. In each case, the mathematical specification of utility flows directly from a mission-specific measure of performance as in (3), and leads directly to a natural partition of state information into private and public components. This partition is evident even without a complete enumeration of every state component: it is imposed by consideration of which terms in the utility function depend on information that must be available to all sensors, and which terms depend on information that is specific to an individual sensor.

The framework we have described extends directly to missions other than the two considered here. This is because mathematical representation of the utility for any mission proceeds directly from the mission-specific measure of performance, from the characteristics of the measurements that are obtained by individual sensing tasks, and from the way these measurements are used to update the representation of the external world. As long as it is possible to define a mission-relevant measure of performance and a reasonable model for the extraction and incorporation of new sensor measurements, it will be possible to derive a

[9] As in the multi-target tracking mission, closely spaced targets or wide-field-of-view sensors may result in a single task assignment providing data on multiple targets. As with that mission, it is possible to accommodate this possibility, but for the sake of simplicity, such accommodation is not done here.

mission-specific utility function and to partition state into public and private components. For instance, consideration of coverage or wide-area search missions, in which the goal is to surveil a continuous region of interest through repeated sensing tasks, leads to straightforward utility functions and state compositions for such missions [9]. Consideration of persistent surveillance missions, in which individual targets or regions must be kept under continuous sensor watch, leads to other utility functions and state compositions specific to those missions. The multi-target tracking and multi-target identification missions examined here provide two important, but by no means exhaustive, examples of the applicability of our general approach.

# 4 Optimization

Utility provides a quantification of the expected mission benefit for a particular sensing task. It provides a means for different potential sensing tasks, and different collections of sensing tasks, to be directly compared. As described in Section 2 and indicated in Figure 2, schedule selection involves optimization over the full set of utilities for all possible sensing tasks, which are provided as auction bids. In this formulation, $M$ sensors each provide bids on $N$ tasks over each of $L$ distinct timesteps (possibly withholding bids on any tasks they deem infeasible). Schedule selection thus consists of selecting some subset of $LMN$ possible sensing tasks that meet various logical or operational constraints on sensor usage. Depending on $L$, $M$, $N$, and the constraints involved, this may be difficult: in fact, in the absence the general case, it is a combinatorially hard problem. Fortunately, it is possible to formulate the required optimization as a mixed integer linear programming (MILP) problem [2]. MILP optimization is a mature field that has been the subject of extensive research and development for decades. It has yielded a variety of robust, efficient solution algorithms that are available in commercial and open-source software packages. As such, formulation of schedule selection as a MILP program has numerous theoretical and practical benefits.

MILP optimization, like linear programming optimization [1], involves the maximization or minimization of a linear function of a collection of variables subject to linear inequality constraints on those variables. Unlike linear programming, MILP requires that the variables subject to optimization take on only integer values. The generic form of a MILP problem is

$$\begin{array}{rl} \max & \mathbf{c}'\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{Z}^K. \end{array} \tag{15}$$

Formulation of sensor scheduling as a MILP problem thus requires possible sensor taskings to be specified in terms of an appropriate integer-valued vector $\mathbf{x}$, the impact of different sensor taskings on a desired outcome to be described in terms of $\mathbf{c}$, and relevant tasking constraints to be indicated in terms of $\mathbf{A}$ and $\mathbf{b}$.

It is straightforward to encapsulate potential sensor taskings in a discrete-valued vector. In fact, the indicator variables $s_{l,m,n}$ introduced in Section 2 provide exactly the needed specification. We can collect the full set of $LMN$ scalar indicator values into a single vector:

$$\mathbf{s} = \left[ \begin{array}{cccccccccc} s_{1,1,1} & s_{1,1,2} & \cdots & s_{1,1,N} & s_{1,2,1} & \cdots & s_{1,M,N} & s_{2,1,1} & \cdots & s_{L,M,N} \end{array} \right]' \tag{16}$$

As indicated in (3), each of the $LMN$ potential sensing tasks corresponds to a particular utility that can be calculated by the sensor at hand. We can collect the full set of $LMN$ such utilities into a single vector whose elements parallel those of (16):

$$\mathbf{u} = \left[ \begin{array}{cccccccccc} u_{1,1,1} & u_{1,1,2} & \cdots & u_{1,1,N} & u_{1,2,1} & \cdots & u_{1,M,N} & u_{2,1,1} & \cdots & u_{L,M,N} \end{array} \right]' \tag{17}$$

Choice of any sensor schedule amounts to selection of a particular $\mathbf{s}$ in which some elements are 1 and some elements are 0. The total utility realized by this $\mathbf{s}$ is the sum of the utilities in $\mathbf{u}$ that correspond to the unity-valued elements of $\mathbf{s}$. This sum is simply the inner product $\mathbf{u}'\mathbf{s}$. Hence, if we can encapsulate relevant constraints on sensor usage in terms of a matrix-product inequality on $\mathbf{s}$, then sensor scheduling becomes a MILP problem of the form

$$\begin{array}{rl} \max & \mathbf{u}'\mathbf{s} \\ \text{s.t.} & \mathbf{A}\mathbf{s} \leq \mathbf{b}, \\ & \mathbf{s} \in \{0,1\}^{LMN} . \end{array} \tag{18}$$

Note that the requirement that the elements of $\mathbf{s}$ be integer-valued in the canonical MILP form of (15) is modified to a requirement in (18) that they be binary. This requirement could easily be enforced in (18) through the construction of specific rows in $\mathbf{A}$ and $\mathbf{b}$. The form of (18) is a special case of the general MILP problem that is known as a $(0,1)$-programming problem [2], and we use it here for the sake of conciseness and clarity.

Encapsulation of relevant sensor tasking constraints in $\mathbf{A}$ and $\mathbf{b}$ amounts to specifying linear inequalities on the collection of $s_{l,m,n}$, each of which equates to a single row of $\mathbf{A}$ and a corresponding element of $\mathbf{b}$. Some sensor-scheduling constraints are logical: for instance, a particular agent will generally only be able to perform one task at a time. Within each timestep $l$, this constraint can be expressed for each agent $m$ as

$$\sum_{n=1}^{N} s_{l,m,n} \leq 1. \tag{19}$$

Each of the set of $LM$ such constraints would be encoded in terms of a row of $\mathbf{A}$ in which exactly $N$ elements have value 1, in which the other $(LMN-N)$ elements have value 0, and in which the corresponding element of $\mathbf{b}$ also has value 1. More generally, if the task indicated by $s_{l,m,n}$ requires $d_{l,m,n}$ timesteps to complete, and if agent $m$ will be completely occupied in performing the indicated task until it is complete, then (19) can be modified to express this constraint as follows:

$$\sum_{n=1}^{N} \sum_{i=l}^{d_{l,m,n}} s_{l+i-1,m,n} \leq 1. \tag{20}$$

Each of the set of $LM$ such constraints, like those of (19), is encodable in terms of a row of $\mathbf{A}$ in which all elements are 0 or 1, and for which the corresponding element of $\mathbf{b}$ is 1.

The formulation of (18) also admits constraints that are operational rather than logical. Simultaneous use of sensors to service multiple missions might dictate that a sensor only be used a limited number of times in a given schedule. This constraint can be expressed for each agent $m$ as

$$\sum_{l=1}^{L} \sum_{n=1}^{N} s_{l,m,n} \leq k \tag{21}$$

for some upper-bound $k$. Each of the set of $M$ such constraints would be encoded in terms of a row of $\mathbf{A}$ in which exactly $L$ elements have value 1, in which the other $(LMN-L)$ elements have value 0, and in which the corresponding element of $\mathbf{b}$ is equal to $k$. Similarly, use of an active sensor such as radar to image a particular target might make simultaneous sensing of that target by any sensor impractical, due to interference concerns. This constraint is expressible for each target $n$ as

$$\sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{i=1}^{d_{l,m,n}} s_{l+i-1,m,n} \leq 1. \tag{22}$$

Each of the set of $N$ such constraints would be encoded in terms of a row of $A$ in which all elements are 0 or 1, and for which the corresponding element of $\mathbf{b}$ is 1.

The richness of the MILP formulation allows for specification of a wide variety of realistic constraints. Encoding these constraints mathematically is usually straightforward, as demonstrated with the previous examples. Translation of these mathematical constraints into rows of $\mathbf{A}$ and $\mathbf{b}$ is often tedious, but is not theoretically difficult [10]. Use of a high-level solver interface or advanced modeling language can facilitate reliable implementation and error-free solution of the sensor-scheduling MILP problem by allowing its specification directly in terms of mathematical equations or high-level software functions, rather than in terms of low-level matrices [7].

# 5 Demonstration

We have implemented a demonstration version of the autonomous multi-platform sensor scheduling system described here [14]. It is implemented in Matlab, and it provides a modeling, simulation, and visualization platform for algorithm demonstration and analysis. It enables simulation of targets and sensing platforms that move either stochastically or according to deterministic trajectories that may be provided via input files. It allows for specification of realistic models for platform dynamics, target motion, sensor dynamics, and sensor performance.

The demonstration version of the autonomous multi-platform sensor scheduling statem contains a full implementation of the multi-target tracking mission described in Section 3.1 and a partial implementation of the multi-target identification mission described in Section 3.2. It is extensible to support other missions through specification and implementation of other utility functions representing those missions' measures of performance, as described in Section 3. The simulation implements strong public and private state partitioning, with information accessibility among different agents limited to that provided by explicit communication. The public ISR state required for the mission implementation at hand is maintained centrally and made available to all agents, while the private state needed by each agent to calculate its own utilities is maintained locally, and is not accessible by any other agents. The simulation contains a full implementation of the auction process specified in Figure 2. The MILP optimization is formulated as described in Section 4 and is performed using the intlinprog function in the Matlab Optimization toolbox.

The demonstration implementation provides for visualization of the sensing platform and target locations as the engagement evolves. It also provides for after-the-fact visualization of relevant system metrics, including the overall measure of performance $P$ and other ancillary quantities. Figure 4 provides two screenshots from the visualization of a tracking-mission engagement involving three sensing agents and ten targets. The tracker in this mission is implemented as a Kalman filter tracker, in which location measurements extracted stochastically by individual sensors from their collected are provided to the auctioneer, which incorporates them in the public ISR-state target location estimates. The visualization depicted in each image in Figure 4 includes the locations of the sensing platforms and targets, as well as the track covariances of each target (shown as ellipses in the ground plane). The top image depicts the beginning of an engagement, before any tasks have been assigned to any sensors; the bottom image depicts a time later in the engagement, after repeated sensing tasks have reduced the track covariances (as is evident from the smaller ground-plane ellipses in the bottom image, compared to the top image). This provides a visual indication of the effectiveness of the auction approach applied to a utility function derived from a mission-specific measure of performance.

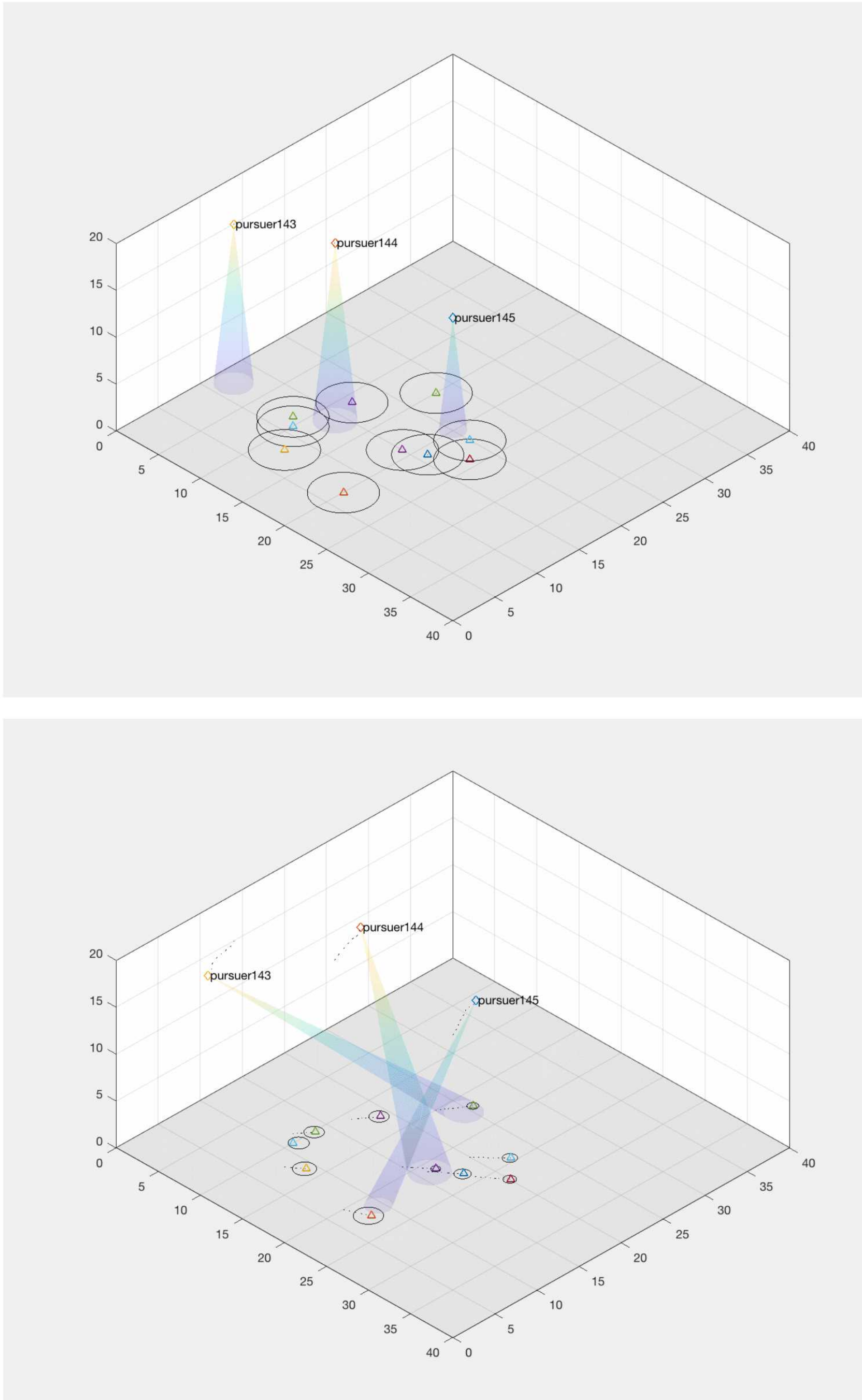


Figure 4: Demonstration-implementation visualizations of tracking mission: start of engagement (top) and later in engagement (bottom).

# References


[1] D. Bertsimas and J.N. Tsisiklis. *Introduction to Linear Optimization*. Athena Scientific, Nashua, NH, 1997.

[2] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer Programming*. Graduate Texts in Mathematics. Springer, New York, 2014.

[3] T.M. Cover and J.A. Thomas. *Elements of Information Theroy*. Wiley, New York, 1991.

[4] C.E. García, D.M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, May 1989.

[5] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[6] B.P. Gerkey and M.J. Matarić. Sold! Auction methods for multirobot coordination. *IEEE Transactions on Robotics*, 18(5):758–768, October 2002.

[7] W.E. Hart, C.D. Laird, J.-P. Watson, D.L. Woodruff, G.A. Hackebeil, B.L. Nicholson, and J.D. Siirola, editors. *Pyomo—Optimization Modeling in Python*. Springer, New York, 2nd edition, 2017.

[8] R.P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699–738, June 1987.

[9] J.A. Richards. Coverage metrics and utility. Technical note. Sandia National Laboratories, Albuquerque, NM, September 20, 2018.

[10] J.A. Richards. An integer programming formulation of multi-sensor task allocation. Technical note. Sandia National Laboratories, Albuquerque, NM, October 30, 2018.

[11] J.A. Richards. A simple probabilistic assessment model. Technical note. Sandia National Laboratories, Albuquerque, NM, February 9, 2018.

[12] J.A. Richards. Target identification entropies and utility. Technical note. Sandia National Laboratories, Albuquerque, NM, March 19, 2018.

[13] J.A. Richards. Track covariances and utility. Technical note. Sandia National Laboratories, Albuquerque, NM, February 6, 2018.

[14] R. Schlossman. Automonous multi-platform sensor scheduling (AMPSS) modeling, simulation, and visualization environment. Technical note. Sandia National Laboratories, Albuquerque, NM, August 2019.