A Geometric Transversal Approach to Analyzing Track Coverage in Sensor Networks

Kelli Baumgartner, *Member*, *IEEE*, and Silvia Ferrari, *Member*, *IEEE*

Abstract—This paper presents a new coverage formulation addressing the quality of service of sensor networks that cooperatively detect targets traversing a region of interest. The problem of track coverage consists of finding the positions of n sensors such that a Lebesgue measure on the set of tracks detected by at least k sensors is optimized. This paper studies the geometric properties of the network, addressing a deterministic track coverage formulation and binary sensor models. It is shown that the tracks detected by a network of heterogeneous omnidirectional sensors are the geometric transversals of nontranslates families of circles. A novel methodology based on cone theory is presented for representing and measuring sets of transversals in closed form. Then, the solution to the track coverage problem can be formulated as a nonlinear program (NLP). The numerical results show that this approach can improve track coverage by up to two orders of magnitude compared to grid and random deployments. Also, it can be used to reduce the number of sensors required to achieve a desired detection performance by up to 50 percent and to optimally replenish or reposition existing sensor networks.

Index Terms—Coverage, sensor networks, target track, search theory, geometric transversals, optimization.

1 Introduction

The study of sensor network deployment for cooperatively detecting moving targets in a region of interest (ROI) has recently received considerable attention [1], [2], [3]. Several authors have pointed out that a fundamental problem is the placement of sensors to provide a desired level of coverage, or quality of service, in the ROI, which in this paper is assumed to be a bounded subset of a two-dimensional euclidean space [4], [5], [6], [7]. The problem of *area coverage* is concerned with subsets of two-dimensional space that lie within the range of at least one sensor in the network [8], [9]. Then, the area coverage of the network is the union of the areas representing the sensors' field of views divided by the area of the ROI [10]. Depending on the geometry of the sensor field of view, this quality of service may be optimized by placing the sensors through circle packing algorithms [11], [12] or by solving integer linear programs that place directional sensors (sectors) to cover multiple known targets [7]. In *point coverage*, a sensing performance function is defined in terms of the distance between the sensor and a point in two-dimensional space and sensors are placed to provide uniform performance over the ROI. Sensor placement for optimal point coverage was performed using Voronoi diagrams in [13]. Another well-known formulation is the *art-gallery problem*, or line-of-sight visibility [14], [15]. This coverage problem aims at placing sensors in an ROI with obstacles such that a set of fixed targets is in the line of sight of at least one sensor. The formulation most closely related to that presented in this paper is *grid coverage* [4], in which sensors are placed such that every point on a grid lies within the detection range of at least k sensors.

The problem of *track coverage* pertains to the ability of a sensor network to cooperatively detect targets traversing the ROI. It is motivated by track-before-detect surveillance systems that employ proximity sensors to establish the presence of passive moving targets over a large ROI, with no a priori knowledge of the target track [16]. In these sensor networks, the objective is to detect target tracks. A track is said to be detected if it can be formed from multiple elementary detections that occur independently at various times [17], [18], [19], [20]. In [17], an event-based algorithm was developed to form the potential track of a target moving in a straight line, based on multiple closest-point-of-approach (CPA) detections obtained by a proximity network. In [18], [19], [20], closed-form solutions for probability of track detection were obtained using search theory and Poisson approximations by assuming a uniform distribution of sensors with constant range and by modeling the moving target as a two-state Markov processes. This paper focuses on the geometric properties of these sensor networks, obtaining a measure of their quality of service as a function of the sensors' positions and ranges. Consequently, the network performance can be optimized by deciding how to place, reposition, or replenish the sensors.

The track coverage problem presented in this paper is to find the positions of n sensors such that a measure of the tracks detected by at least k sensors in a rectangular ROI is optimized. This paper assumes binary omnidirectional sensor models [4], [10], [14] and straight target tracks [17], [18], [19], [20]. The result is a new geometric transversal

- *K. Baumgartner is with the Mission Analysis Branch, Analex Corporation, a subsidiary of QinetiQ North America, 1000 Apollo Dr., Brook Park, OH 44142. E-mail: Kelli.Baumgartner@analexcleveland.com.*
- *S. Ferrari is with the Pratt School of Engineering, Laboratory for Intelligent Systems and Controls (LISC), Duke University, Durham, NC 27708-0005. E-mail: sferrari@duke.edu.*

Manuscript received 2 Apr. 2007; revised 10 Jan. 2008; accepted 28 Jan. 2008; published online 28 Mar. 2008.
Recommended for acceptance by Y.-M. Wang.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2007-04-0109.
Digital Object Identifier no. 10.1109/TC.2008.56.

problem that requires a closed-form representation of the lines intersecting nontranslates families of circles, also known as stabbers [21]. In this paper, a new methodology is presented for representing these stabbers by means of cones that are finitely generated by pairs of unit vectors (Section 4). Then, the quality of service of the network is expressed by a coverage function obtained by assigning a Lebesgue measure to the cones of stabbers (Section 5). In Section 6, this function is used to derive the probability of track detection and to formulate the track coverage problem as a nonlinear program (NLP).

The track coverage problem is complementary to the path exposure problem. Path exposure is the ability of a sensor network to observe a target moving along a specified path [22], [23]. In [24], a random sequential sensor deployment scheme was developed using the worst and best-case exposure of paths in an ROI. Although exposure-based deployment schemes do not optimize sensor positions, they are very useful for addressing the detection of active targets that may maneuver to avoid the sensors. On the other hand, when targets are passive, sensors positioned by optimizing track coverage (through NLP) always outperform those positioned by other deployment schemes (Section 7). In Section 7.1, the NLP approach is shown to improve track coverage by up to two orders of magnitude compared to the sequential and grid deployment presented in [4] and [24]. As shown in Section 7.2, the NLP approach employs significantly smaller networks than path-exposure deployment [24] (e.g., 50 percent fewer sensors) to achieve a desired probability of track detection. In Sections 7.3 and 7.4, the NLP approach is used to improve the track coverage of an existing network by up to 69.4 percent by either replenishing or repositioning the sensors.

2 Problem Formulation and Assumptions

The track coverage problem consists of placing a set of omnidirectional sensors in an ROI such that a measure of the set of tracks that are cooperatively detected is maximized. The present formulation is in two-dimensional euclidean space and relies on the following assumptions:

1. targets move along straight paths,
2. the ROI is a rectangle $\mathcal{A}$,
3. the field of view of each sensor can be represented by a circle centered at the sensor location, and
4. a sensor may detect a target only if the target track intersects its field of view.

Suppose a network of n proximity sensors with different ranges $r_1, \ldots, r_n$ must be deployed in the ROI for the purpose of detecting moving targets. Then, the number of detections required per track is a constant parameter k such that $1 \leq k \leq n$ and its value is decided based on the level of confidence required by the sensor system [17]. For example, Fig. 1 illustrates how two CPA detections may be used to form four possible tracks (the two in the figure and their reflections) in track-before-detect surveillance systems [17], [18], [19], [20].

Therefore, in this paper, we address the following problem:

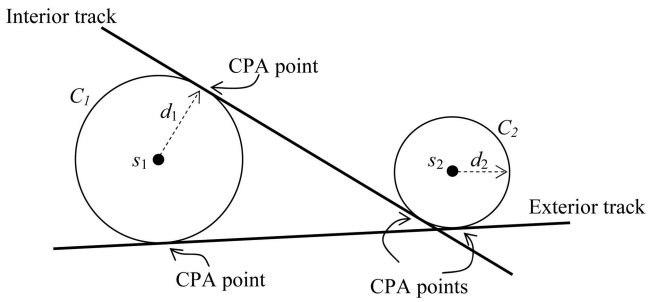

Fig. 1. Geometry of interior and exterior tracks formed from two CPA detections obtained by two omnidirectional sensors placed at s_1 and s_2 (adapted from [17], reflections are omitted for simplicity).

Problem 2.1 (Track Coverage Optimization). *Given a parameter $1 \leq k \leq n$ and a network S of n omnidirectional sensors with ranges $R_S = \{r_1, \ldots, r_n\}$, find the sensor positions $X_S = \{s_1, \ldots, s_n\}$ inside a rectangular ROI $\mathcal{A}$ such that a measure of the set of tracks detected by at least k sensors in S is maximized.*

In order to optimize the sensors' placement, the set of tracks they intercept is expressed as a function of the sensors' coordinates in the plane $s_1, \ldots, s_n$. Under the given assumptions, track coverage can be viewed as a new geometric transversal problem. In this paper, a novel approach is presented for representing geometric transversals by means of cones.

3 Background on Geometric Transversals

A set of geometric objects in $\mathbb{R}^d$ is said to have a j-*transversal* when the objects are simultaneously intersected by a j-dimensional flat or translate of a linear space. A line transversal $(j = 1)$, also referred to as a *stabber*, with $d = 2$ and $k \geq 1$, is a straight line that intersects at least k members of a family of objects. For example, line transversals of a family of five square polygons, with $k = 3$, are shown in Fig. 2. Geometric transversal theory is concerned with the analysis of the space of transversals to a family of compact convex bodies in $\mathbb{R}^d$ [25]. While considerable attention has been given to establishing the necessary and sufficient conditions for the existence of transversals, algorithms for finding j-transversals or for constructing a space of transversals have been obtained in only a few special cases [21]. It has been shown that the problem of finding point transversals for families of half spaces is a linear program [26]. The problem of finding j-transversals to a family of n polytopes in $\mathbb{R}^d$ can be formulated in terms of a system of linear matrix inequalities. One of the first line transversal algorithms constructing the stabbing region for a family of n line segments was developed in [27] to address the practical visibility problem in the plane. This algorithm has been extended to a family of simple convex sets whose boundaries intersect pairwise at most s times [28], thereby finding line transversals for homothets of simple planar objects [29] or for circles of equal radius [30].

One of the results that is relevant to the track coverage problem is an algebraic decision tree methodology that

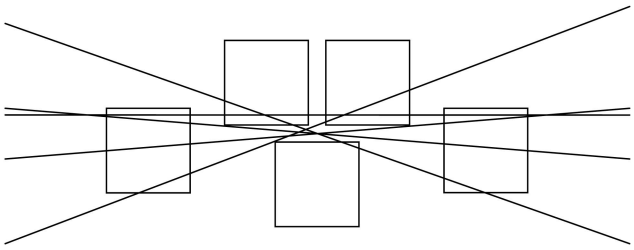

Fig. 2. Examples of line transversals for a family of $n = 5$ square polygons and $k = 3$ (taken from [21, p. 182]).

finds a single line transversal for a translates family of n line segments in $\mathbb{R}^2$ or n equal circles in $\mathbb{R}^2$ [31]. Although existing algorithms (e.g., [31]) cannot construct closed-form representations of line transversals, they could be applied to determine the tracks intercepted by a family of omnidirectional sensors with known positions. It was pointed out in [21] that geometric transversal algorithms could be greatly improved by considering the underlying geometric nature of the problem. In this paper, the geometric properties of circles and cones are used to construct efficient closed-form representations of line transversals. Then, a Lebesgue measure is assigned to the space of line transversals to obtain a track coverage function in terms of variable sensors' positions (Section 5).

4 Cone Representation of Track Coverage

Based on the problem formulation in Section 2, the tracks detected by k sensors in an omnidirectional network S of size n can be viewed as the line transversals of a family of circles with different radii. We show that a set of line transversals can be represented by means of a coverage cone, which contains line transversals (or tracks) characterized by the same intercept. The coverage cone of a single sensor is defined in the following section. In Section 4.2, we obtain the coverage cone of multiple sensors. Then, the track coverage over a rectangular area is represented by the union of coverage cones with intercepts along the perimeter (Section 4.3). Finally, the coverage cone representation of line transversals is used in Section 5 to obtain a track coverage function that quantifies the ability of a sensor network to perform cooperative detections.

4.1 Coverage Cone

Consider a sensor in the network S that is indexed by i and is located at $s_i = [x_i\ y_i]^T \in \mathbb{R}^2$ in the xy-plane. Let $C(s_i, r_i) = C_i$ denote a circle with radius r_i centered at s_i that represents the field of view of sensor i. Assume that any target track can be described by a straight line, $y = a_y x + b_y$, with slope a_y and y-intercept b_y. As shown in [17], a CPA detection event takes place when the target path is tangential to a circle of radius $d_i \leq r_i$, centered at s_i. Without loss of generality, we can assume that all circles and CPA detections are in the positive orthant $\mathbb{R}^2_+$. Then, we can represent tracks by rays or half lines denoted by $\mathcal{R}_\alpha(b_y)$. Each ray originates at an intercept $y = b_y$ and forms an angle $\alpha = \tan^{-1}(a_y)$ with the x-axis. Let the vector $y_0 \equiv [0\ b_y]^T$ denote the position of the y-intercept. Then, the position of the ith sensor can be expressed by a relative position vector that is convenient for generating the sensor coverage cone, namely,

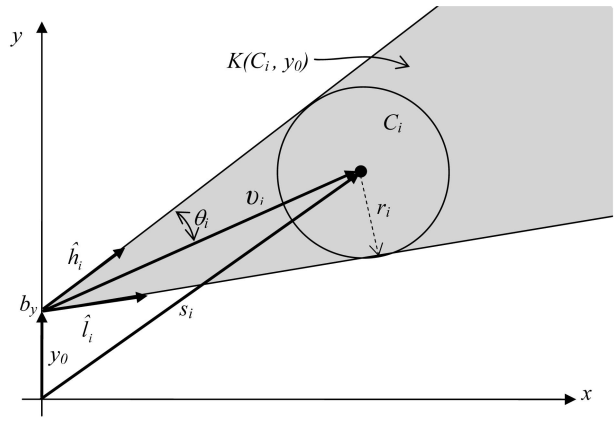

Fig. 3. Coverage cone $K(C_i, y_0)$ of a sensor located at s_i, generated by the unit vectors $\hat{l}_i$ and $\hat{h}_i$.

$$v_i \equiv (s_i - y_0) = \begin{bmatrix} x_i \\ (y_i - b_y) \end{bmatrix}. \tag{1}$$

Borrowing two basic definitions from convex analysis [32], a set K is said to be a *cone* if, for all $x \in K$, where $x \in \mathbb{R}^2$ and $c > 0$, we have $cx \in K$. Also, given a nonempty subset X of $\mathbb{R}^n$, the *cone generated by* X is the set of all nonnegative combinations of the elements of X, denoted by $\mathrm{cone}(X)$. We define the *coverage cone* of the ith sensor with respect to the intercept b_y to be the cone generated by C_i with origin y_0 and we denote it by $K(C_i, y_0)$. The coverage cone is a basic construct for the coverage function because it represents the set of tracks that can be detected by the ith sensor.

Remark 4.1. The coverage cone $K(C_i, y_0)$ contains the set of all tracks $\mathcal{R}_\alpha(b_y)$ that intersect the sensor field of view $C_i(s_i, r_i)$ in $\mathbb{R}^2_+$.

The proof is provided in Appendix B and an example of a coverage cone is illustrated in Fig. 3.

Let θ_i denote half the opening angle of the coverage cone (Fig. 3). Since the extremals of K are tangential to C_i, the trigonometric relationships

$$\sin\theta_i = \frac{r_i}{\|v_i\|} = \frac{r_i}{\sqrt{x_i^2 + (y_i - b_y)^2}} \tag{2}$$

and

$$\cos\theta_i = \frac{\sqrt{\|v_i\|^2 - (r_i)^2}}{\|v_i\|} \tag{3}$$

relate the opening angle to the sensor location s_i through v_i in (1). Then, the coverage cone $K \subset \mathbb{R}^2$ is finitely generated by two unit vectors $\hat{l}_i$ and $\hat{h}_i$, that is,

$$K(C_i, y_0) = \mathrm{cone}(\hat{l}_i, \hat{h}_i) = \{x \mid x = c_1\hat{l}_i + c_2\hat{h}_i,\ c_1, c_2 \geq 0\}, \tag{4}$$

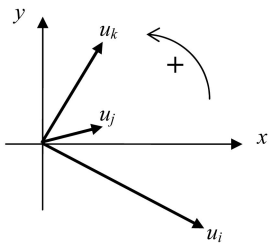

Fig. 4. Example of three vectors ordered according to the xy-frame, where $u_i \prec u_j \prec u_k$.

provided the unit vectors are obtained from v_i through rotation matrices

$$\hat{h}_i = \begin{bmatrix} \cos\lambda_i \\ \sin\lambda_i \end{bmatrix} = Q_i^+ \hat{v}_i \equiv \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \frac{v_i}{\|v_i\|} \quad (5)$$

and

$$\hat{l}_i = \begin{bmatrix} \cos\gamma_i \\ \sin\gamma_i \end{bmatrix} = Q_i^- \hat{v}_i \equiv \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \frac{v_i}{\|v_i\|}, \quad (6)$$

where $Q_i^- = (Q_i^+)^T$. Thus, the coverage cone $K(C_i, y_0)$ is completely specified by the unit vectors $\hat{l}_i$ and $\hat{h}_i$, which are known functions of s_i and r_i.

4.2 k-Coverage Cone for Multiple Sensors

Multiple sensor detections typically are necessary to determine target tracks by means of proximity sensors or in the presence of measurement errors and false alarms, as shown in [17]. Let k denote the minimum number of distinct sensor detections that are required by the system to reliably form a track. Two detections are said to be distinct when they are obtained by two different sensors. Thus, k detections are obtained when k sensors in the network $S = \{C_1, \ldots, C_n\}$ intersect the same track.

In this section, we show that the set of tracks that intersect at least k sensors in S, with y-intercept b_y, is contained by a so-called k-coverage cone. Vectors in $\mathbb{R}^2$ are ordered according to the orientation of the reference frame. Two vectors u_i and u_j are said to be *ordered* according to the xy-frame such that $u_i \prec u_j$ if, when these vectors are translated to make their origins coincide and u_i is rotated through the smallest angle possible to meet u_j, this rotation is in the same direction as the orientation of the xy-frame (as illustrated in Fig. 4 and in [33]). Let $\Omega(S, y_0)$ and $\Lambda(S, y_0)$ denote the sets of unit vectors generating the coverage cones of all sensors in S with origin y_0. That is, from (5) and (6), $\Omega(S, y_0) = \{\hat{h}_i | Q_i^- \hat{h}_i = \hat{v}_i, \forall i \in I_S\}$ and $\Lambda(S, y_0) = \{\hat{l}_i | Q_i^+ \hat{l}_i = \hat{v}_i, \forall i \in I_S\}$, where I_S denotes the index set of S. Then, these two sets can be used to determine the k-coverage cone of S, as shown by the following result:

Proposition 4.2. *The set of all tracks $\mathcal{R}_\alpha(b_y)$ that are line transversals to a family of k nontranslates circles $\{C_1, \ldots, C_k\} \equiv S_k$ with index set I_{S_k} is contained by the finitely generated cone*

$$K_k(S_k, y_0) = \text{cone}(\hat{l}^*, \hat{h}^*), \quad (7)$$

where $y_0 = [0 \quad b_y]^T$, $\hat{h}^* = \hat{h}_\jmath$, and $\hat{l}^* = \hat{l}_\imath$ with $\jmath, \imath \in I_{S_k}$ such that $\hat{h}_\jmath \preceq \hat{h}_i \in \Omega(S_k, y_0)$ and $\hat{l}_\imath \succeq \hat{l}_i \in \Lambda(S_k, y_0)$ for $\forall i \in I_{S_k}$ and provided $\hat{l}_\imath \prec \hat{h}_\jmath$. If $\hat{l}_\imath \succ \hat{h}_\jmath$, then $K_k(S_k, y_0) = \emptyset$. A proof is provided in Appendix C.

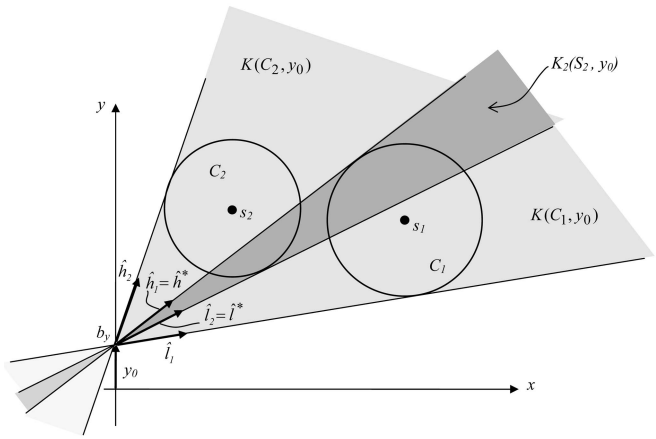

Fig. 5. The k-coverage cone $K_2(S_2, y_0)$ of the family $S_2 = \{C_1, C_2\}$ is shown in dark gray and is generated by the unit vectors $\hat{l}^*$ and $\hat{h}^*$ obtained from the sets of unit vectors generating $K(C_1, y_0)$ and $K(C_2, y_0)$ (shown in light gray).

A simple example of the k-coverage cone is illustrated in Fig. 5, where $k = 2$ and S_2 contains two sensors located at s_1 and s_2. In this example, the 2-coverage cone $K_2(S_2, y_0)$ is generated by the unit vectors $\hat{l}^* = \hat{l}_2$ and $\hat{h}^* = \hat{h}_1$ since $\Omega(S_2, y_0) = \{\hat{h}_1, \hat{h}_2\}$ and $\Lambda(S_2, y_0) = \{\hat{l}_1, \hat{l}_2\}$, where $\hat{l}_2 \succ \hat{l}_1$ and $\hat{h}_1 \prec \hat{h}_2$.

The cone $K_k(S_k, y_0)$ is referred to as the k-coverage cone of S_k with origin y_0. An important feature of this approach is that the k-coverage cone is easily obtained from the sets of unit vectors Ω and Λ. Provided two unit vectors are in the first or fourth quadrant of a reference frame, they can be ordered by their direction sines (as shown in Appendix D). Therefore, if we let

$$\begin{aligned} \sin\lambda^* &= \inf\{\sin\lambda_i | \hat{h}_i = [\cos\lambda_i \ \ \sin\lambda_i]^T \in \Omega(S_k, y_0), \ \forall i \in I_{S_k}\}, \\ \sin\gamma^* &= \sup\{\sin\gamma_i | \hat{l}_i = [\cos\gamma_i \ \ \sin\gamma_i]^T \in \Lambda(S_k, y_0), \ \forall i \in I_{S_k}\}, \end{aligned} \quad (8)$$

then $\hat{l}^* = [\cos\gamma^* \ \ \sin\gamma^*]^T$ and $\hat{h}^* = [\cos\lambda^* \ \ \sin\lambda^*]^T$. When the unit vectors are in the second or third quadrant, they can still be ordered by their direction sines by introducing a constant rotation (Appendix D). Therefore, the infimum and supremum in (8) can be determined by linear operations on the elements of Ω and Λ, respectively.

Consider now the tracks detected by at least k sensors in $S = \{C_1, \ldots, C_n\}$, with $1 \le k < n$. These tracks are the line transversals of any k-subset of S. A k-subset is defined as a subset containing any k elements of a set with n elements [34]. By Proposition 4.2, all tracks $\mathcal{R}_\alpha(b_y)$ detected by a set of k sensors S_k are contained by the k-coverage cone of S_k. It follows that the set of all tracks $\mathcal{R}_\alpha(b_y)$ detected by at least k sensors in S is the union of the k-coverage cones of all k-subsets of S:

$$\mathcal{K}_k(S, y_0) = \bigcup_{j=1}^{m} K_k(S_k^j, y_0), \quad m = \binom{n}{k}. \quad (9)$$

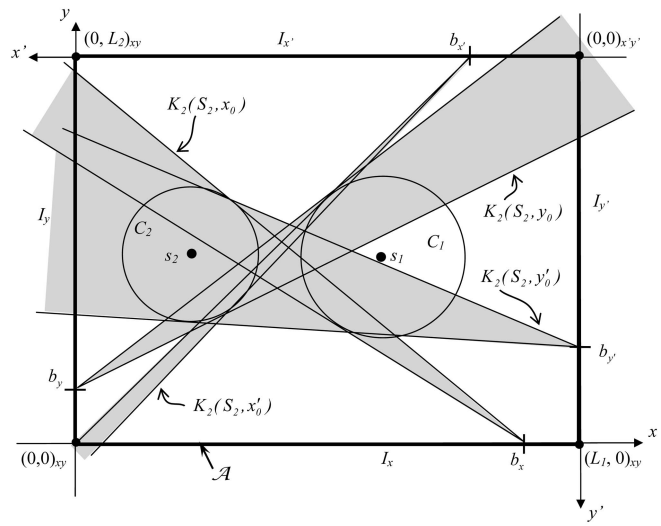

Fig. 6. Reference frames used to define k-coverage cones with respect to each axis, as illustrated in the figure for $k = 2$ and $S_2 = \{C_1, C_2\}$.

S_k^j denotes the jth k-subset of S and the number m of possible k-subsets is given by the binomial coefficient n *choose* k, as shown in (9). Since $\mathcal{K}_k$ is a union of possibly disjoint cones, it may not be a cone [32]. Nevertheless, a measure defined for a cone can be applied to $\mathcal{K}_k$ using the principle of inclusion-exclusion, as shown in Section 5. In the next section, the k-coverage cone is used to construct an approximate representation of the set of tracks that traverse the ROI $\mathcal{A}$, and are detected by at least k sensors in S.

4.3 Track Coverage over a Rectangular Region of Interest (ROI)

Let the ROI be a rectangle $\mathcal{A}$ of known dimensions $L_1 \times L_2$. Place the xy-frame of reference along two sides of $\mathcal{A}$ such that its origin $(0,0)_{xy}$ coincides with one vertex and one side of $\mathcal{A}$ can be denoted by the interval $\mathcal{I}_y \equiv \{y | y \in [0 \quad L_2]\}$ (Fig. 6). The set of all tracks $\mathcal{R}_\alpha(b_y)$ that intersect this side of $\mathcal{A}$ at $b_y \in \mathcal{I}_y$ and are detected by at least k sensors in S is $\mathcal{K}_k(S, y_0)$ in (9). In order to obtain representations that are computationally tractable, $\mathcal{I}_y$ is discretized into N_2 increments of size $\delta b = L_2/N_2$. Letting $b_y^\ell \equiv \ell \cdot \delta b$ and $y_0^\ell \equiv [0 \quad b_y^\ell]^T$, the set of tracks that intersect $\mathcal{I}_y$ and are detected by at least k sensors in S can be approximated by

$$\mathcal{K}_k(S, \mathcal{I}_y) \approx \bigcup_{\ell=0,\ldots,N_2} \mathcal{K}_k(S, y_0^\ell), \quad N_2 = L_2/\delta b, \tag{10}$$

where each set $\mathcal{K}_k(S, y_0^\ell)$ is given by (9). Clearly, by letting $\delta b \to 0$, the above approximation approaches the entire set of tracks intersecting $\mathcal{I}_y$.

The methodology is extended to all sides of $\mathcal{A}$ by placing a second frame of reference, $x'y'$, along the remaining sides of $\mathcal{A}$ such that its origin $(0,0)_{x'y'}$ is the vertex opposite to $(0,0)_{xy}$ (Fig. 6). Then, each side of $\mathcal{A}$ is denoted by one of the intervals $\mathcal{I}_y$, $\mathcal{I}_x \equiv \{x | x \in [0 \quad L_1]\}$, $\mathcal{I}_{x'} \equiv \{x' | x' \in [0 \quad L_1]\}$, or $\mathcal{I}_{y'} \equiv \{y' | y' \in [0 \quad L_2]\}$. With this choice of reference frames, an efficient representation of the target-tracks traversing $\mathcal{A}$ can be obtained by defining coverage cones with origins on each of the four axes, namely, $x_0 = [b_x \quad 0]^T$, $y_0' = [0 \quad b_{y'}]^T$, and $x_0' = [b_{x'} \quad 0]^T$, where $b_x \in \mathcal{I}_x$, $b_{x'} \in \mathcal{I}_{x'}$, and $b_{y'} \in \mathcal{I}_{y'}$ (Fig. 6). The coverage cones of the ith sensor with respect to each axis are denoted by $K(C_i, y_0)$, $K(C_i, x_0)$, $K(C_i, y_0')$, and $K(C_i, x_0')$ and are obtained by defining a relative-position vector for each axis. From hereon, denote the vector in (1) by $v_i(y_0)$ and let $v_i(x_0) = (s_i - x_0)$ denote the relative-position vector for x. The relative-position vectors for the x' and y' axes are defined as

$$v_i(x_0') = L - s_i - x_0' \text{ and } v_i(y_0') = L - s_i - y_0', \text{ for } \forall i \in I_S, \tag{11}$$

where $L \equiv [L_1 \quad L_2]^T$. The coordinate transformation $s_i|_{x'y'} = (L - s_i)$ is used to express all sensor positions with respect to the same coordinate frame xy. Then, the results in Sections 4.1 through 4.2 can be extended to all axes.

For simplicity, all intervals $\mathcal{I}_y$, $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$ are discretized by increments of the same size δb. Hence, from (10), the set of tracks traversing $\mathcal{A}$ and intersecting at least k sensors in S is given by

$$\begin{aligned}\mathcal{K}_k(S, \mathcal{A}) &= \mathcal{K}_k(S, \mathcal{I}_y) \cup \mathcal{K}_k(S, \mathcal{I}_x) \cup \mathcal{K}_k(S, \mathcal{I}_{x'}) \cup \mathcal{K}_k(S, \mathcal{I}_{y'}) \\ &\approx \left(\cup_{\ell=0}^{N_2} \cup_{j=1}^{m} K_k(S_k^j, y_0^\ell) \cup K_k(S_k^j, y_0'^\ell) \right) \\ &\cup \left(\cup_{\ell=0}^{N_1} \cup_{j=1}^{m} K_k(S_k^j, x_0^\ell) \cup K_k(S_k^j, x_0'^\ell) \right),\end{aligned} \tag{12}$$

where m is equal to the binomial coefficient k *choose* n (as in (9)), $N_2 = L_2/\delta b$, and $N_1 = L_1/\delta b$.

4.4 Example: Assessing the Track Coverage of a Known Sensor Network Configuration with $n = 20$ and $k = 3$

In this example, the cone representation of track coverage is verified by considering a sensor network S, with $n = 20$, $k = 3$, and the known ranges and positions in Fig. 7a. The union of k-coverage cones $\mathcal{K}_k(S, y_0^\ell)$, with $y_0^\ell = [0 \quad 15]^T$, is illustrated in Fig. 7a. The cone representation of track coverage, $\mathcal{K}_k(S, \mathcal{A})$, is computed using the methodology in Section 4.3 and plotted in parameter space in Fig. 7b, where gray represents sets of tracks detected by at least k sensors. When the sensors' positions are known, a subset of tracks that are cooperatively detected by S can be obtained numerically by testing the intersections of a designated sample of tracks $T_\mathcal{R}$ with S, using the inequality

$$d_i = \left| \frac{(b_y + a_y x_i - y_i)}{\sqrt{a_y^2 + 1}} \right| \leq r_i, \tag{13}$$

where $a_y = \tan(\alpha)$ and r_i, x_i, and y_i are the known range and coordinates of sensor i. A proof and derivation are provided in [35]. Let B_i denote a logical array or truth table in which every element corresponds to one track in $T_\mathcal{R}$ and is either equal to 1 or 0, depending on whether the track has been detected (1) or missed (0) by the ith sensor. Every element of B_i can be evaluated using (13) and an array B_i can be obtained for every sensor in S. Then, the logical array

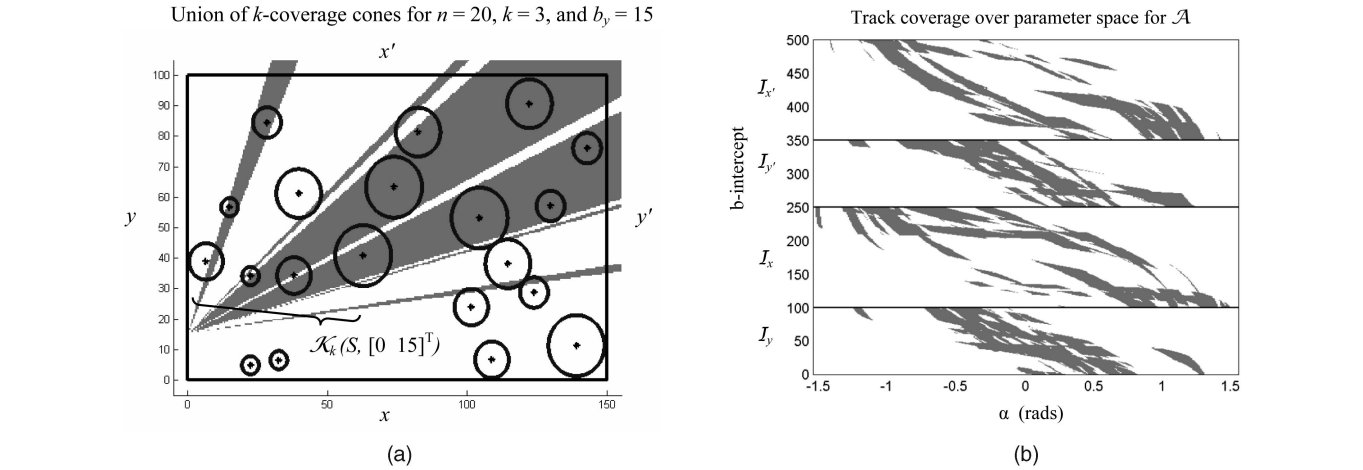

Fig. 7. Track coverage $\mathcal{K}_k(S, \mathcal{A})$ (b) of a known sensor network configuration (a) with $n = 20$, $k = 3$. The union $\mathcal{K}_k(S, y_0^\ell)$ is illustrated by the gray cones in (a) for $y_0^\ell = [0\ 15]^T$.

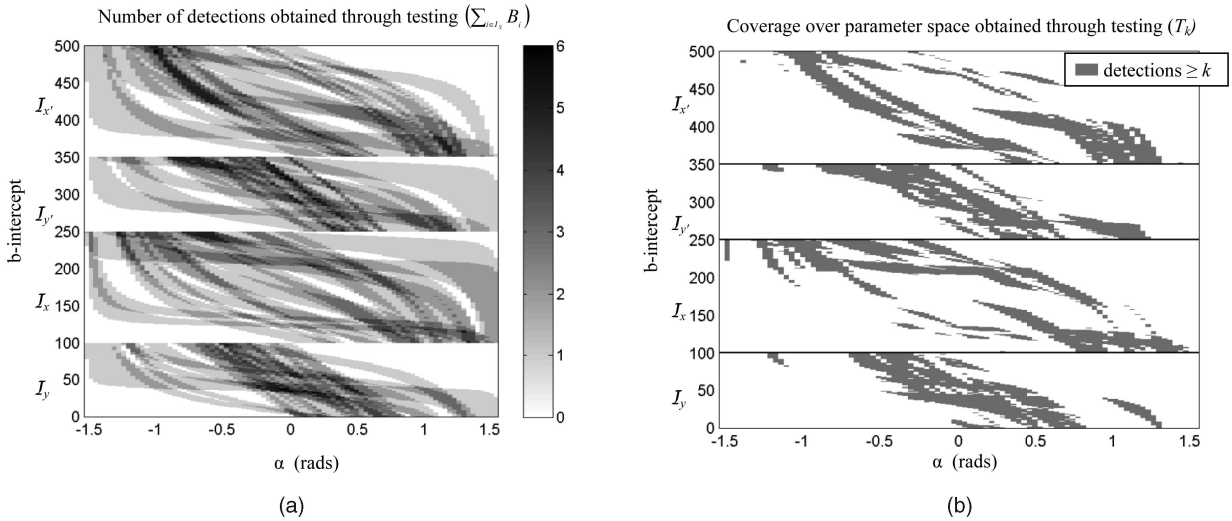

Fig. 8. (a) Number of detections obtained through testing and (b) resulting track coverage for the sensor network in Fig. 7a.

$$T_k = \left\{ \sum_{i \in I_S} B_i \geq k \right\} \tag{14}$$

indicates whether each track in $T_{\mathcal{R}}$ has been detected by at least k sensors in S. The array T_k is computed for the sensor network in Fig. 7a and plotted in parameter space in Fig. 8b. By comparing Fig. 7b to Fig. 8, it can be seen that $\mathcal{K}_k(S, \mathcal{A})$ provides an accurate representation of the tracks that are cooperatively detected by S.

5 Track Coverage Function

The cone representation of track coverage allows us to generate the space of tracks that are cooperatively detected by a sensor network using sets of unit vectors. Another important use of coverage cones is the functional representation of the quality of service of the network. Assign a Lebesgue measure μ on $[0, \pi]$ to any set of rays $K \subset \mathbb{R}^2$ such that $\mu\{\alpha : \mathcal{R}_\alpha \in K\}$. Then, the opening angle of the cone K is a measure on the set of rays contained by K. It follows from Remark 4.1 that the opening angle of the coverage cone $K(C_i, y_0)$ is a measure on the set of tracks through y_0 that are detected by the sensor C_i. Similarly, it follows from Proposition 4.2 that the opening angle of the k-coverage cone $K_k(S_k, y_0)$ is a measure on the set of tracks through y_0 that are detected by all sensors in S_k.

Based on Section 4.2, it is always possible to generate the k-coverage cone of a set S_k by means of two unit vectors $\hat{l}^*$ and $\hat{h}^*$. This unit vector representation also allows us to compute the opening angle of *any* coverage cone by means of the cross product. Let $\psi = \psi(S_k, y_0)$ denote the opening angle of the k-coverage cone $K_k(S_k, y_0)$ in (7), with origin $y_0 = [0 \quad b_y]^T$ and $b_y \in \mathcal{I}_y$. This cone is finitely generated by two unit vectors, $\hat{l}^* = \hat{l}_i$ and $\hat{h}^* = \hat{h}_j$, that are defined in terms of the relative-position vector (1). From the properties of the cross product

$$\sin \psi = \|\hat{l}^* \times \hat{h}^*\|. \tag{15}$$

Thus, using (5) and (6), the opening angle can be written with respect to the sensors' positions as

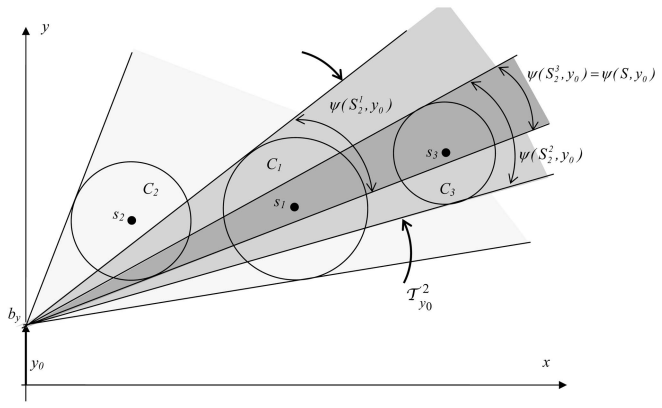

Fig. 9. An example of coverage function, $\mathcal{T}^2_{y_0}$, for three sensors $S = \{C_1, C_2, C_3\}$ located at $X_S = \{s_1, s_2, s_3\}$ and $k = 2$.

$$\psi = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \tag{16}$$

where

$$M_{\imath\jmath} \equiv \begin{bmatrix} \hat{l}^{*T} \\ \hat{h}^{*T} \end{bmatrix} = \begin{bmatrix} (\hat{v}_\imath)^T Q_\imath^+ \\ (\hat{v}_\jmath)^T Q_\jmath^- \end{bmatrix}, \ \hat{v}_i \equiv \frac{(s_i - y_0)}{\|(s_i - y_0)\|} \text{ for } i = \imath, \jmath. \tag{17}$$

$H[\cdot]$ denotes the Heaviside function and $\det(\cdot)$ denotes the matrix determinant. From Proposition 4.2, $\imath$ and $\jmath$ are the indices of the unit vectors $\hat{l}_\imath$ and $\hat{h}_\jmath$ with $\imath, \jmath \in I_{S_k}$ such that $\hat{l}_\imath \succeq \hat{l}_i \in \Lambda(S_k, y_0)$ and $\hat{h}_\jmath \preceq \hat{h}_i \in \Omega(S_k, y_0)$ for $\forall i \in I_{S_k}$ (obtained as shown in Appendix D). The Heaviside function in (16) ensures that, if $\hat{l}_\imath \succ \hat{h}_\jmath$, then $\psi = 0$.

Consider now the case in which $1 \leq k \leq n$. Since the set in (9) is not always a cone, the Lebesgue measure μ on $\mathcal{K}_k(S, y_0)$ is obtained using the principle of inclusion-exclusion [36], as shown by the following result:

Theorem 5.1. *A measure on the set $\mathcal{K}_k(S, y_0)$ for a family of nontranslates circles $S = \{C_1, \ldots, C_n\} \subset \mathbb{R}^2_+$ is given by*

$$\mathcal{T}^k_{y_0}(X_S) = \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \psi(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, y_0),$$
$$m = \frac{n!}{(n-k)!k!}, \tag{18}$$

where the summation $\sum_{1 \leq i_1 < \ldots < i_j \leq m}$ is a sum over all the $[m!/(m-j)!\, j!]$ distinct integer j-tuples $(i_1, \ldots, i_j)$ satisfying $1 \leq i_1 < \ldots < i_j \leq m$. $S_k^{i_l}$ denotes the i_lth k-subset of S and the union $S_k^{i_1} \cup \ldots \cup S_k^{i_j}$ is a p-subset of S, with $k \leq p \leq n$.

A proof is provided in Appendix E. In the remainder of this paper, the union $S_k^{i_1} \cup \ldots \cup S_k^{i_j}$ is abbreviated as $S_p^{i_{1,j}}$.

As an example, consider a network $S = \{C_1, C_2, C_3\}$, with $k = 2$ and positions shown in Fig. 9. From (18), a measure of the set of tracks detected by at least two sensors in S is

$$\begin{aligned} \mathcal{T}^2_{y_0} = {} & \psi(S_2^1, y_0) + \psi(S_2^2, y_0) + \psi(S_2^3, y_0) \\ & - [\psi(S_2^1 \cup S_2^2, y_0) + \psi(S_2^1 \cup S_2^3, y_0) \\ & + \psi(S_2^2 \cup S_2^3), y_0] + \psi(S_2^1 \cup S_2^2 \cup S_2^3, y_0), \end{aligned} \tag{19}$$

where, from the definition of k-subset: $S_2^1 = \{C_1, C_2\}$, $S_2^2 = \{C_1, C_3\}$, and $S_2^3 = \{C_2, C_3\}$. But, the union of two or more k-subsets of S always produces a p-subset of S, with $k < p \leq n$. In this case, $S_2^1 \cup S_2^2 = \{C_1, C_2, C_3\} = S$ and $S_2^1 \cup S_2^3 = S_2^2 \cup S_2^3 = S_2^1 \cup S_2^2 \cup S_2^3 = \{C_1, C_2, C_3\} = S$. Therefore, (19) simplifies to

$$\mathcal{T}^2_{y_0} = \psi(S_2^1, y_0) + \psi(S_2^2, y_0) + \psi(S_2^3, y_0) - 2\psi(S, y_0), \tag{20}$$

as illustrated in Fig. 9.

$\mathcal{T}^k_{y_0}$ provides a measure of the set of tracks through y_0 that are detected by at least k sensors in S as a function of X_S. It is evaluated by summing the opening angles of the coverage cones of all p-subsets of S, with $k \leq p \leq n$. For a p-subset S_p, the coverage cone $K_p(S_p, y_0)$ is generated by two unit vectors, according to Proposition 4.2, and its opening angle $\psi(S_p, y_0)$ is given by the cross product in (16). The above result is used to derive a coverage function for $\mathcal{A}$ (Section 5.1) which can be optimized with respect to X_S, as shown in Section 6.

5.1 ROI Track Coverage Function

The track coverage function for a rectangular ROI, $\mathcal{A}$, is obtained by considering the sets of tracks intersecting its four sides, $\mathcal{I}_y$, $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$, and leading to at least k detections by S. These sets can be represented by coverage cones, as illustrated in Section 4.3. Consider the set of tracks that intersect $\mathcal{I}_y$ and are detected by at least k sensors, $\mathcal{K}_k(S, \mathcal{I}_y)$, in (10). The sets in (10) are all disjoint because they contain rays with different intercepts, thus $\mathcal{K}_k(S, y_0^{\ell_i}) \cap \mathcal{K}_k(S, y_0^{\ell_j}) = \emptyset$ when $\ell_i \neq \ell_j$. Using the definition of Lebesgue measure for disjoint sets [37], it follows that a measure of the set $\mathcal{K}_k(S, \mathcal{I}_y)$, obtained from (10) and (18), is

$$\begin{aligned} \mathcal{T}^k_{\mathcal{I}_y}(X_S) &= \sum_{\ell=0}^{N_2} \mathcal{T}^k_{y_0^\ell}(X_S) \\ &= \sum_{\ell=0}^{N_2} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \ldots < i_j \leq m} \psi(S_p^{i_{1,j}}, y_0^\ell), \end{aligned} \tag{21}$$

where m, $S_p^{i_{1,j}}$, and the j-tuples $(i_1, \ldots, i_j)$ are all defined as in Theorem 5.1.

The sets $\mathcal{K}_k(S, \mathcal{I}_y)$, $\mathcal{K}_k(S, \mathcal{I}_x)$, $\mathcal{K}_k(S, \mathcal{I}_{x'})$, and $\mathcal{K}_k(S, \mathcal{I}_{y'})$ in (12) are not disjoint because a track intersecting one side of $\mathcal{A}$ always intersects one other side of $\mathcal{A}$. Let the opening angles of the k-coverage cones $K_k(S_k, x_0)$, $K_k(S_k, y_0')$, and $K_k(S_k, x_0')$ be denoted by $\zeta(S_k, x_0)$, $\xi(S_k, y_0')$, and $\rho(S_k, x_0')$, respectively. Then, the measure μ on $\mathcal{K}_k(S, \mathcal{A})$ is

$$\mathcal{T}_{\mathcal{A}}^{k}(X_S) = \frac{1}{2}[\mathcal{T}_{\mathcal{I}_y}^{k}(X_S) + \mathcal{T}_{\mathcal{I}_x}^{k}(X_S) + \mathcal{T}_{\mathcal{I}_{x'}}^{k}(X_S) + \mathcal{T}_{\mathcal{I}_{y'}}^{k}(X_S)]$$
$$= \frac{1}{2}\sum_{\ell=1}^{N_2}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}[\psi(S_p^{i_{1,j}}, y_0^{\ell}) + \xi(S_p^{i_{1,j}}, y_0'^{\ell})]$$
$$+ \frac{1}{2}\sum_{\ell=0}^{N_1-1}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}[\zeta(S_p^{i_{1,j}}, x_0^{\ell}) + \rho(S_p^{i_{1,j}}, x_0'^{\ell})], \tag{22}$$

where each term $\cup_{j=1}^{m} K_k(S_k, \cdot)$ in (12) has been written in terms of opening angles using Theorem 5.1.

In the following sections, the track coverage function is used to optimize the deployment of sensor networks performing cooperative target detection. Also, the coverage cones and their opening angles are used to derive an upper bound for the track coverage function and the probability of cooperative target detection of the network.

6 Track Coverage Optimization and Probability of Track Detection

The objective of the track coverage problem is to place a set of sensors in an ROI such that their ability to cooperatively detect moving targets is optimized. Using the track coverage function obtained in Section 5.1, Problem 2.1 can be formulated as an NLP. As shown in [16], [17], [18], [19], [20], in order to obtain multiple independent CPA detections, track-before-detect surveillance systems typically require sensors to lie in $\mathcal{A}$ without overlapping. Then, the set of optimal sensor positions X_S^* is given by the solution $\{s_1^*, \ldots, s_n^*\}$ of the following NLP:

$$\text{maximize } \mathcal{T}_{\mathcal{A}}^{k}(X_S), \tag{23}$$

$$\text{subject to } (x_i - x_j)^2 + (y_i - y_j)^2 > (r_i + r_j)^2,\ \forall i, j \in I_S, \tag{24}$$

$$0 < x_i < L_1,\ \forall i \in I_S, \tag{25}$$

$$0 < y_i < L_2,\ \forall i \in I_S, \tag{26}$$

where $s_i^* = [x_i^* \quad y_i^*]^T$ and the objective function $\mathcal{T}_{\mathcal{A}}^{k}(X_S)$ is given by (22). Also, the NLP (23)-(26) can be easily modified to add sensors optimally to an existing network. Suppose f sensors already exist in $\mathcal{A}$ and there is an opportunity for replenishing the network with q additional sensors. The NLP (23)-(26) can be written for a network $S = \{C_1, \ldots, C_f, C_{f+1}, \ldots, C_n\}$ with $n = q + f$ sensors, where now $\{s_1, \ldots, s_f\}$ are known constants and $\{s_{f+1}, \ldots, s_n\}$ are the NLP variables. Then, its solution $\{s_{f+1}^*, \ldots, s_n^*\}$ represents the set of sensor positions for optimally replenishing the network.

It is shown in Appendix G that the track coverage function (22) has an upper bound

$$\mathcal{T}_{\mathcal{A}}^{max} = \left(\frac{L_1 + L_2}{\delta b}\right)\pi \ge \mathcal{T}_{\mathcal{A}}^{k}(X_S), \quad \text{for} \quad \forall\, X_S, k, n, \tag{27}$$

that is independent of k and n. This upper bound represents the track coverage provided by a sensor network that detects all tracks through $\mathcal{A}$ at least k times, where $\mathcal{A}$ is $L_1 \times L_2$. Therefore, it is referred to as *total track coverage*. In large sensor networks, total track coverage may be achieved by concentric configurations placed around the perimeter of $\mathcal{A}$. However, in many applications, the available sensors are not sufficient to provide total track coverage. Thus, $\mathcal{T}_{\mathcal{A}}^{k}$ is maximized by determining the optimal placement X_S^* from (23)-(26).

The coverage-cone representation of track coverage is also used to derive the probability of detection of targets in $\mathcal{A}$ as a function of X_S. In applications where there is no prior knowledge of target tracks, any ray $\mathcal{R}_{\alpha}(b_y)$ has the same probability of representing an actual target track. Then, the probability that a target traversing $\mathcal{A}$ along a straight path is detected by at least k sensors in S is

$$P_{\mathcal{A}}^{k}(X_S) = \frac{\delta b}{4\pi L_2}\sum_{\ell=1}^{N_2}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}[\psi(S_p^{i_{1,j}}, y_0^{\ell}) + \xi(S_p^{i_{1,j}}, y_0'^{\ell})]$$
$$+ \frac{\delta b}{4\pi L_1}\sum_{\ell=0}^{N_1-1}\sum_{j=1}^{m}(-1)^{j+1}\sum_{1\le i_1<\ldots<i_j\le m}[\zeta(S_p^{i_{1,j}}, x_0^{\ell}) + \rho(S_p^{i_{1,j}}, x_0'^{\ell})], \tag{28}$$

where m, $S_p^{i_{1,j}}$, and the j-tuples $(i_1, \ldots, i_j)$ are defined as in Theorem 5.1. A proof is provided in Appendix H. As in the previous sections, the opening angles ψ, ξ, ζ, and ρ are given by the functions in Appendix F and are computed for every coverage cone of the p-subsets in (28), with $k \le p \le n$. The derivation in Appendix H can be modified to account for nonuniform probabilities of the tracks' heading and intercept, as will be shown in a separate paper.

7 Results

A number of sensor deployment problems can be formulated as an NLP optimizing the track coverage function (22). Several algorithms are available for solving NLPs numerically [38], [39]. In this research, two NLP software packages [40], [41] were implemented and it was found that the MATLAB Optimization Toolbox *fmincon* function, based on Sequential Quadratic Programming (SQP) [41], performs satisfactorily for the track coverage optimization problems presented in this section. In every simulation, the NLP solution, X_S^*, is determined by using multiple random initializations to avoid local maxima. Although up to 100 initializations are used, X_S^* is typically determined after approximately 20 trials, when the optimum track coverage ceases to improve. All trials were found to converge to a local maxima without encountering cycles [40] or other degenerate behaviors.

In Section 7.1, the NLP approach is shown to improve track coverage by up to two orders of magnitude compared to sequential and grid deployment [4], [24]. Also, a greedy algorithm implementing the track coverage function (22) is found to be considerably more effective than random or grid deployment. In Section 7.2, the NLP solution is used to deploy sensors until a desired probability of track detection,

TABLE 1
Sensor Networks Size and Range

n	R_S (Km)
10	{ 3, 3, 5, 5, 6, 6, 8, 8, 10, 10 }
15	{ 3, 3, 3, 5, 5, 5, 6, 6, 6, 8, 8, 8, 10, 10, 10 }
20	{ 3, 3, 3, 3, 5, 5, 5, 5, 6, 6, 6, 6, 8, 8, 8, 8, 10, 10, 10, 10}
40	{ 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 8, 8, 8, 8, 8, 8, 8)}

TABLE 2
Normalized Track Coverage as a Function of Network Parameters and Deployment Strategy

Network Parameters	$\mathcal{T}_{\mathcal{A}}^{k}(X_S)/\mathcal{T}_{\mathcal{A}}^{max}$ (NLP Improvement %)			
n, k	NLP	Random	Grid	Greedy
10,2	0.304	0.169 (79.9%)	0.158 (92.4%)	0.300 (1.33%)
10,3	0.158	0.033 (379%)	0.0392 (303%)	0.151 (4.64%)
10,4	0.0700	3.00×10^{-3} (2.23×10^{3}%)	4.90×10^{-4} (1.42×10^{4}%)	0.0680 (2.94%)
15,3	0.286	0.0764 (274%)	0.0912 (214%)	0.250 (14.4%)
15,4	0.172	0.0179 (861%)	0.0117 (1.37×10^{3}%)	0.149 (15.4%)
20,3	0.364	0.183 (98.9%)	0.169 (115%)	0.325 (12.0%)
40,3	0.578	0.440 (31.4%)	0.450 (28.4%)	0.471 (22.7%)
40,4	0.423	0.202 (109%)	0.226 (87.2%)	0.354 (19.5%)

(28), is achieved, requiring significantly smaller networks than path-exposure deployment [24]. Finally, in Sections 7.3 and 7.4, the NLP approach is used to improve the track coverage of an existing network by either replenishing or repositioning the sensors, respectively.

7.1 Track Coverage Optimization

The effectiveness of the NLP solution to the track coverage optimization problem (Problem 2.1) is demonstrated for the sensor networks in Table 1. The number of required detections, k, is made to vary between 2 and 4 and $\mathcal{A}$ has dimensions $L_1 \times L_2 = 150 \times 100$ km. Also, a fast and effective greedy algorithm implementing the track coverage function, Algorithm 1, is obtained by modifying the circle-packing algorithm presented in [11], which places circles in a rectangle one at a time based on heuristic criteria and on their maximum hole degree performance [11]. The heuristic criteria are that the first circle is placed in the bottom-left corner of $\mathcal{A}$ and each subsequent circle must border two items (one side of $\mathcal{A}$ or another circle) and avoid overlapping. It is found that, by implementing the track coverage function (22) in lieu of the maximum hole degree performance function [11], the resulting deployment is considerably more effective than grid and random deployments.

Algorithm 1: Pseudocode of greedy track coverage algorithm

order n sensors in S according to decreasing radii;
place first sensor in bottom-left corner of $\mathcal{A}$;
for $i = 2$ to n **do**
  generate all eligible positions for sensor C_i;
  Require: C_i touches two items;
  for (every eligible placement of C_i) **do**
    calculate the coverage $\mathcal{T}_{\mathcal{A}}^{k}(\{s_1, \ldots, s_{i-1}, s_i\})$;
  end for
  select eligible placement s_i with maximum track coverage;
end for

Table 2 compares the track coverage of sensors placed at the NLP solution X_S^*, to the greedy Algorithm 1, and to random and grid deployments. $\mathcal{T}_{\mathcal{A}}^{k}(X_S)$ is normalized by $\mathcal{T}_{\mathcal{A}}^{max}$ to enable the comparison between different sensor networks and parameters. These results show that sensors placed at the NLP solution provide a track coverage up to 15 times higher (Table 2, $n = 15$ and $k = 3$) than grid and random deployments, which have been previously proposed for cooperative target detection [4], [24]. Sensor networks deployed by the greedy algorithm display a track coverage within 1.3 percent-22.7 percent of the optimal $\mathcal{T}_{\mathcal{A}}^{k}(X_S^*)$ (Table 2). Thus, Algorithm 1 may be used in lieu of SQP when computation time is a concern. The NLP, greedy, grid, and random deployments are plotted in Figs. 10 and 11, for $n = 40$ and $k = 3$. The greedy algorithm clusters sensors providing near-optimal track coverage for networks with low area coverage, but otherwise causing track coverage holes. The plot of track coverage in Fig. 12 illustrates that, in this example, the NLP and grid

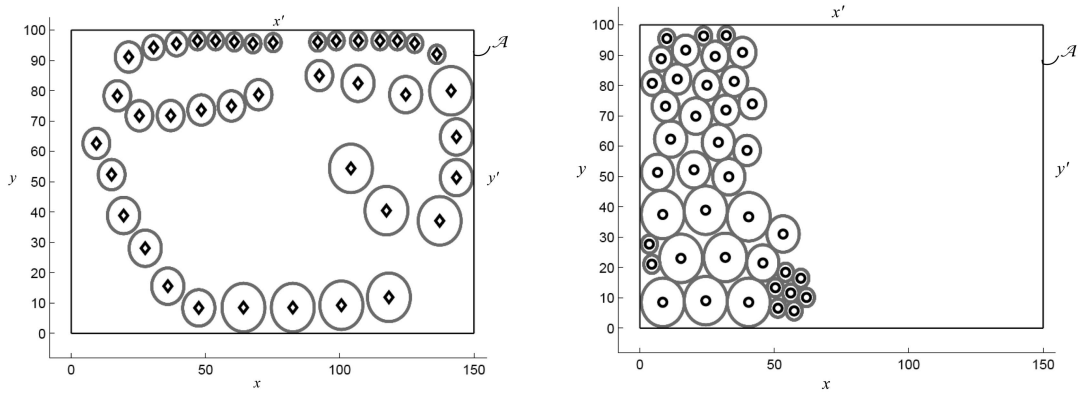

Fig. 10. Deployment of a sensor network with $n = 40$ and $k = 3$ obtained by (a) the NLP solution ($\diamond$) and by (b) the greedy algorithm ($\circ$).

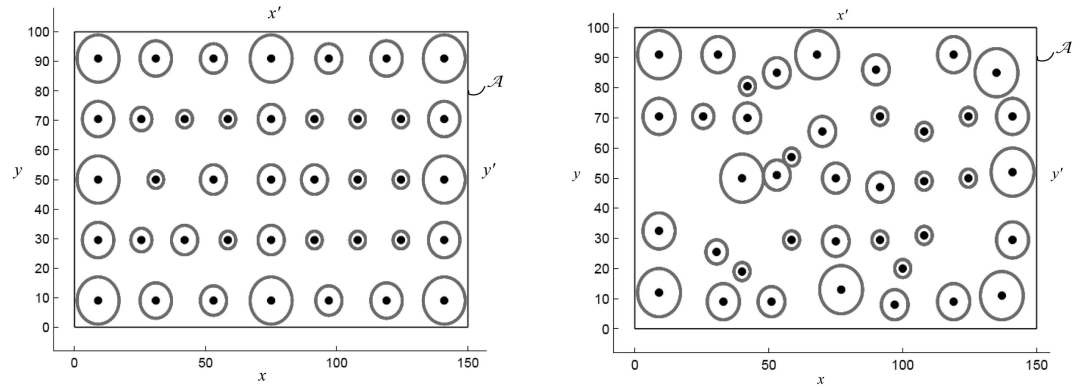

Fig. 11. (a) Grid and (b) random deployments for a sensor network with $n = 40$ and $k = 3$.

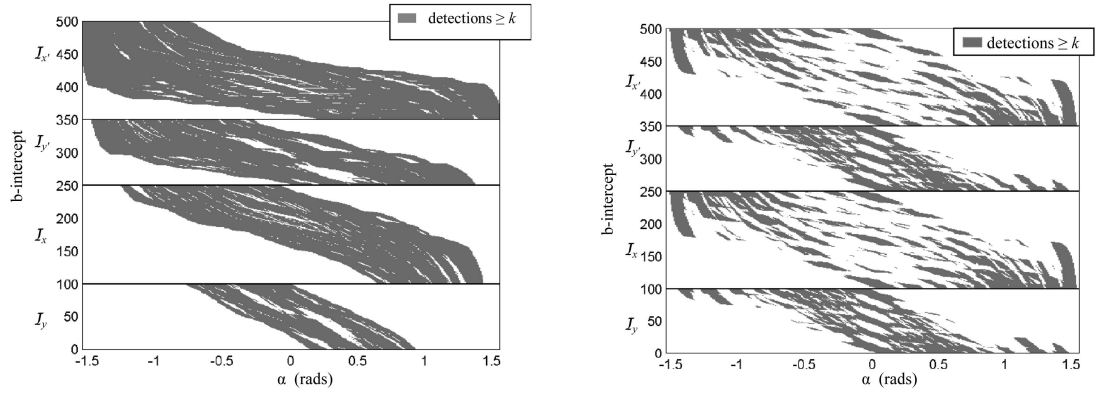

Fig. 12. Track coverage $\mathcal{K}_k(S, \mathcal{A})$ of a sensor network with $n = 40$ and $k = 4$, deployed by (a) NLP and (b) grid strategies.

deployments perform detections in similar regions of parameter space, but the NLP deployment displays far fewer coverage holes, leading to an 87.2 percent increase in track coverage (Table 2).

7.2 Sensor Deployment for Achieving a Desired Probability of Track Detection

A sequential deployment algorithm based on path exposure was presented in [24] to achieve a desired probability of track detection by using a minimal number of sensors. In this section, we implement the sequential algorithm [24] and show that the NLP approach can reduce the number of sensors, $\hat{n}$, required to achieve a desired probability of track detection, $\hat{P}^k_{\mathcal{A}}$, by up to 50 percent. In the first example, all sensors have range $r_i = 5$ km and $\hat{P}^3_{\mathcal{A}} = 0.41$. As shown in Fig. 13a, the sequential algorithm from [24] requires a minimal number of sensors $\hat{n}_{\text{SEQ}} = 40$, whereas the NLP solution requires only $\hat{n}_{\text{NLP}} = 30$ sensors (Fig. 13b). In the second example, the desired probability of track detection is $\hat{P}^3_{\mathcal{A}} = 0.18$ and the size of the network is increased according to the ranges in Table 1. In this case, the sequential algorithm from [24] requires $\hat{n}_{\text{SEQ}} = 20$ sensors,

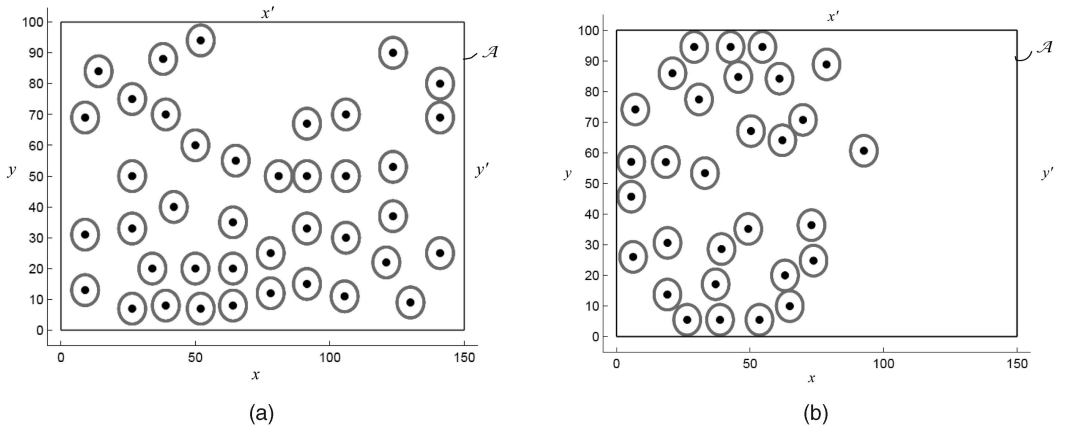

Fig. 13. (a) Sequential deployment of $\hat{n}_{\mathrm{SEQ}} = 40$ sensors and (b) optimal deployment of $\hat{n}_{\mathrm{NLP}} = 30$ sensors for achieving $\hat{P}^3_{\mathcal{A}} = 0.41$ when $r_i = 5$ km for $\forall i$.

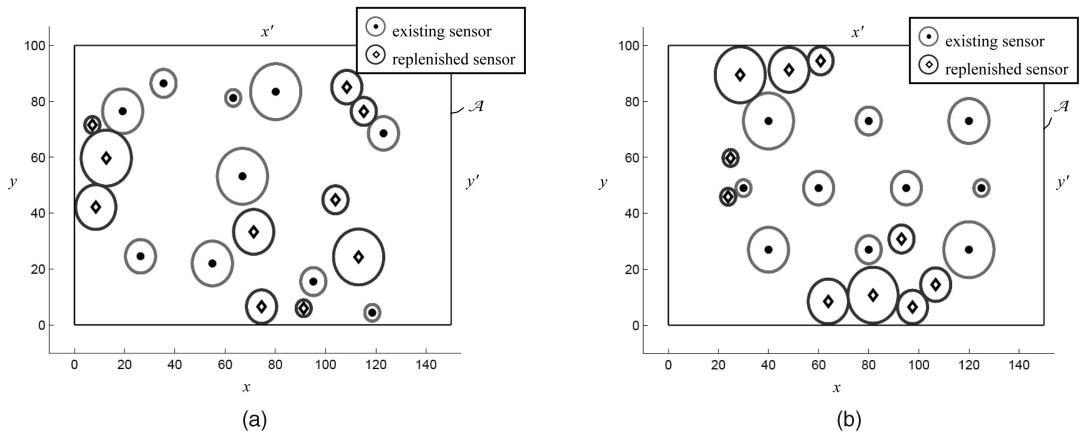

Fig. 14. Optimal replenishment of an existing sensor network with $f = 10$ sensors ($\bullet$) in (a) a random or (b) a grid configuration, $k = 3$, and $q = 10$ replenished sensors ($\diamond$).

whereas the NLP deployment achieves the desired probability of track detection with only $\hat{n}_{\mathrm{NLP}} = 10$ sensors (i.e., 50 percent less than the sequential algorithm).

7.3 Optimal Replenishment of Sensor Networks

In this section, the NLP (23)-(26) is used to deploy a set of sensors for the purpose of replenishing an existing network that performs suboptimally. This situation may come about when sensors have been initially misplaced or have moved due to their environment [16], [19], [42]. It is assumed that the positions of f existing sensors in $\mathcal{A}$ are known and their track coverage must be improved by adding an additional set of q sensors. As an example, consider the sensor network shown by dots in Fig. 14a, with $f = 10$, $k = 3$, and $\mathcal{T}^3_{\mathcal{A}}/\mathcal{T}^{max}_{\mathcal{A}} = 0.033$. When an additional $q = 10$ sensors are placed at the solution X^*_S of the NLP (23)-(26), shown by the diamonds in Fig. 14a, the track coverage of the entire network (with $n = q + f$) is improved by 715.2 percent. On the other hand, if the same q sensors are added using a sequential strategy (adapted from [24]), the track coverage is only improved by 35.9 percent. In another example, the existing sensor network is in a grid configuration, as shown by the dots in Fig. 14b, and provides $\mathcal{T}^3_{\mathcal{A}}/\mathcal{T}^{max}_{\mathcal{A}} = 0.039$. When an additional $q = 10$ sensors are deployed by the NLP algorithm (as shown by diamonds in Fig. 14b), the track coverage is improved by 635.9 percent. If the same sensors are added by sequential deployment [24], the track coverage is only improved by 47.2 percent. Thus, by replenishing a sensor network with the methodology presented in this paper, its track coverage is improved significantly compared to existing deployment schemes.

7.4 Optimal Repositioning of Sensor Networks

In applications where sensors are maneuverable [43], [44], [45], an optimal deployment strategy can be obtained by including the allowed repositioning region in the NLP constraints. Without loss of generality, all sensors are assumed to have the same repositioning capabilities and w is used to denote half the width of a square region within which each sensor can maneuver with the available power (Fig. 15a). Then, the NLP (23)-(26) is modified by replacing the constraints (25)-(26) with the following inequalities:

$$x_i - w < x_i < x_i + w, \ \forall i \in I_S, \tag{29}$$

$$x_i - w > 0, \ \forall i \in I_S, \tag{30}$$

$$x_i + w < L_1, \ \forall i \in I_S, \tag{31}$$

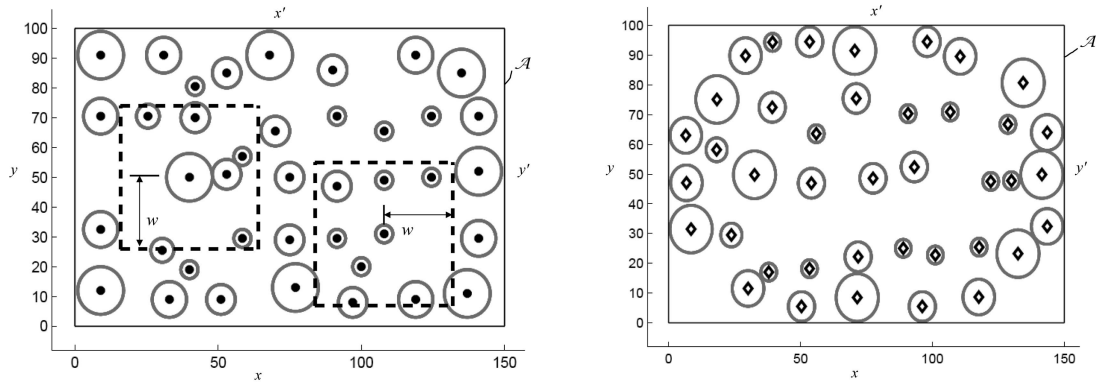

Fig. 15. A suboptimal sensor network with $n = 40$ and a maneuvering region with $w = 24$ km (dashed line) in (a) is optimally repositioned using NLP in (b).

$$y_i - w < y_i < y_i + w, \ \forall i \in I_S, \tag{32}$$

$$y_i - w > 0, \ \forall i \in I_S, \tag{33}$$

$$y_i + w < L_2, \ \forall i \in I_S. \tag{34}$$

The solution X_S^* of the resulting NLP constitutes the new positions to be assumed by the maneuvering sensors in order to improve the overall track coverage of the sensor network. Consider the sensor network in Fig. 15a, with suboptimal track coverage $\mathcal{T}_{\mathcal{A}}^3/\mathcal{T}_{\mathcal{A}}^{max} = 0.44$, $n = 40$, and $w = 24$ km. When these sensors are repositioned at X_S^*, shown by the diamonds in Fig. 15b, the track coverage of the network is improved by 27.7 percent. As another example, consider the sensor network illustrated in Section 4.4, Fig. 7a, with suboptimal track coverage $\mathcal{T}_{\mathcal{A}}^3/\mathcal{T}_{\mathcal{A}}^{max} = 0.183$, $n = 20$, and $w = 24$ km. When this network is repositioned using the above NLP, track coverage is improved by 69.4 percent. Thus, the methodology presented in this paper can be used to improve the track coverage of a sensor network by allowing existing sensors in $\mathcal{A}$ to maneuver subject to power and energy constraints.

8 Summary and Conclusions

This paper presents a novel track coverage formulation addressing the quality of service of sensor networks performing cooperative target detection. In many surveillance applications, simple (e.g., proximity) sensor networks are employed to detect the tracks of passive unauthorized targets in an ROI through multiple elementary detections in an approach known as track-before-detect. This paper investigates the geometric properties of these networks and formulates optimal deployment strategies for cooperative sensor detection using geometric transversals. A novel methodology is presented for representing sets of geometric transversals and a Lebesgue measure on these sets in closed form. Consequently, track coverage can be optimized by solving an NLP. The numerical results show that NLP deployment can increase track coverage by up to two orders of magnitude compared to existing grid and random deployment schemes. Also, it can decrease the number of sensors required to provide a desired probability of track detection by up to 50 percent compared to existing path exposure techniques. Finally, it can significantly improve track coverage by replenishing or repositioning an existing sensor network that displays suboptimal performance due to errors in its initial deployment or to sensors being displaced by their environment.

Appendix A
Nomenclature

- $\mathcal{A}$: Rectangular ROI
- $C_i = C(s_i, r_i)$: Circle with radius r_i centered at s_i
- $\mathrm{cone}(X)$: Cone generated by X
- δb: Size of discretization increments
- γ_i: Angle formed by $\hat{l}_i$ with the x-axis
- $\hat{h}_i$: Upper unit vector generating $K(C_i, y_0)$
- $\hat{h}^*$: Upper unit vector generating $K_k(S_k, y_0)$
- I_S: Index set of S
- $K(C_i, y_0)$: Cone generated by C_i with origin y_0
- $K_k(S_k, y_0)$: k-Coverage cone of S_k with origin y_0
- $\mathcal{K}_k(S, \mathcal{A})$: Set of tracks traversing $\mathcal{A}$ and intersecting at least k sensors in S
- $\mathcal{K}_k(S, y_0)$: Set of tracks through y_0 and intersecting at least k sensors in S
- $\hat{l}_i$: Lower unit vector generating $K(C_i, y_0)$
- $\hat{l}^*$: Lower unit vector generating $K_k(S_k, y_0)$
- L_1 and L_2: Width and height of $\mathcal{A}$
- λ_i: Angle formed by $\hat{h}_i$ with the x-axis
- $\Lambda(S, y_0)$: Set of lower unit vectors generating the coverage cones of all circles in S with origin y_0
- $\Omega(S, y_0)$: Set of upper unit vectors generating the coverage cones of all circles in S with origin y_0
- $P_{\mathcal{A}}^k(X_S)$: Probability of track detection
- $\psi(S_k, y_0)$: Opening angle of $K_k(S_k, y_0)$
- Q_i^- and Q_i^+: Clockwise and counterclockwise rotation matrices, respectively
- r_i and s_i: Range and position of sensor i, respectively
- $\mathcal{R}_\alpha(b_y)$: Ray with y-intercept b_y and slope $\tan\alpha$
- $\rho(S_k, y_0')$: Opening angle of $K_k(S_k, y_0')$
- S: Set of n sensors

- S_k: Set of k sensors
- S_k^j: jth k-subset of S
- $\mathcal{T}_{y_0}^k(X_S)$: Measure of the set of tracks in $\mathcal{K}_k(S, y_0)$
- $\mathcal{T}_{\mathcal{A}}^k(X_S)$: Measure of the set of tracks in $\mathcal{K}_k(S, \mathcal{A})$
- θ_i: Half the opening angle of $K(C_i, y_0)$
- v_i: Relative-position vector of sensor i
- $\hat{v}_i$: Unit relative-position vector of sensor i
- $\xi(S_k, x_0')$: Opening angle of $K_k(S_k, x_0')$
- $\zeta(S_k, x_0)$: Opening angle of $K_k(S_k, x_0)$

APPENDIX B

PROOF OF REMARK 4.1

Let $\mathcal{R}_\alpha(b_y)$ denote a ray that intersects $C_i = C_i(s_i, r_i)$ in $\mathbb{R}_+^2$. Consider any two points that lie on $\mathcal{R}_\alpha(b_y)$ and inside $C_i(s_i, r_i)$ and let $u_1, u_2 \in \mathbb{R}_+^2$ denote their positions relative to the origin y_0 of the coverage cone $K(C_i, y_0)$. By construction, $u_1, u_2 \in C_i(s_i, r_i)$ and a vector z joining the two points will lie on the ray $\mathcal{R}_\alpha(b_y)$. Let c_1 and c_2 denote any two positive constants. By definition of vector sum and subtraction [33], if $z = c_1u_1 + c_2u_2$, then z has the same origin as u_1 and u_2. Thus, since z lies on $\mathcal{R}_\alpha(b_y)$, $\mathcal{R}_\alpha(b_y)$ intercepts the y-axis at the cone origin y_0. If $z = \pm c_1u_1 \mp c_2u_2$, z does not have the same origin as u_1 and u_2 and, thus, $\mathcal{R}_\alpha(b_y)$ does not intercept the y-axis at y_0. By definition, $K(C_i, y_0)$ is the set of all nonnegative combinations of the elements in C_i. Since u_1 and u_2 are two elements in C_i, and any nonnegative combination of these two elements can be written as $c_1u_1 + c_2u_2$, with $c_1, c_2 > 0$, it follows that $z = c_1u_1 + c_2u_2 \in K(C_i, y_0)$. Finally, since $\mathcal{R}_\alpha(b_y)$ denotes any ray with intercept b_y that intersects $C_i = C_i(s_i, r_i)$ in $\mathbb{R}_+^2$ and $z = c_1u_1 + c_2u_2$ provided $\mathcal{R}_\alpha(b_y)$ intercepts the y-axis at y_0, it also follows that any $\mathcal{R}_\alpha(b_y)$ that intersects C_i and the y-axis at y_0 is contained by $K(C_i, y_0)$. □

APPENDIX C

PROOF OF PROPOSITION 4.2

This proof considers a family of $k = 3$ nontranslates $S_3 = \{C_i, C_j, C_l\}$ with index set $I_{S_3} = \{i, j, l\}$. The results can be extended to higher k by induction. From Remark 4.1, a coverage cone $K(C_\ell, y_0)$ contains the set of all tracks $\mathcal{R}(b_y)$ that intersect C_ℓ in $\mathbb{R}_+^2$, where $\ell \in I_{S_3}$. Then, from set theory, the set of tracks intersecting all circles in the family S_3 is given by the following intersection:

$$\begin{aligned} K_3(S_3, y_0) &= \bigcap_{\ell \in I_{S_3}} K(C_\ell, y_0) \\ &= K(C_i, y_0) \cap K(C_j, y_0) \cap K(C_l, y_0). \end{aligned} \tag{35}$$

From the properties of cones [32, p. 70], the intersection of a collection of cones is also a cone. Thus, $K_3(S_3, y_0)$ is a cone. A vector z representing a ray $\mathcal{R}$ lies in a cone K if and only if $\mathcal{R}$ lies in K since any point on $\mathcal{R}$ can be written as cz, with $c > 0$.

Consider any ray $\mathcal{R}^\ell \in K(C_\ell, y_0)$, where $K(C_\ell, y_0) = \text{cone}(\hat{l}_\ell, \hat{h}_\ell)$ and, thus, can be represented by a vector $z_\ell = c_1\hat{l}_\ell + c_2\hat{h}_\ell$ with constants $c_1, c_2 > 0$. Then, $z_\ell \in K(C_\ell, y_0)$ and, by the properties of vector sum, $\hat{l}_\ell \prec z_\ell \prec \hat{h}_\ell$. Next, consider a cone $K^* = \text{cone}(\hat{l}^*, \hat{h}^*)$ that is finitely generated by two unit vectors $\hat{h}^* = \hat{h}_j$ and $\hat{l}^* = \hat{l}_\imath$ with $\jmath, \imath \in I_{S_3}$ and assume $\hat{l}_\imath \prec \hat{h}_j$. By the properties of finitely generated cones [32], any vector $z^* = b_1\hat{l}^* + b_2\hat{h}^*$ with constants $b_1, b_2 > 0$ must lie in K^*. It follows that a ray $\mathcal{R}^*$ with the same slope and origin as z^* must also lie in K^* since any point on $\mathcal{R}^*$ can be written as cz^* with $c > 0$. Since z^* is a positive combination of $\hat{l}^*$ and $\hat{h}^*$, it also follows that $\hat{l}^* \prec z^* \prec \hat{h}^*$.

According to Proposition 4.2, choose $\hat{h}^* = \hat{h}_j \preceq \hat{h}_\ell$ and $\hat{l}^* = \hat{l}_\imath \succeq \hat{l}_\ell$ for $\forall \ell \in I_{S_3}$. Suppose the unit vectors of S_3 can be ordered as $\hat{h}_l \prec \hat{h}_j \prec \hat{h}_i$ and $\hat{l}_i \prec \hat{l}_l \prec \hat{l}_j$. Then, the unit vectors and z^* can be ordered as follows:

$$\hat{l}_\ell \preceq \hat{l}_j = \hat{l}^* \prec z^* \prec \hat{h}^* = \hat{h}_l \preceq \hat{h}_\ell \text{ for } \forall \ell \in \{i, j, l\} = I_{S_3}, \tag{36}$$

or, more explicitly,

$$\hat{l}_i \prec \hat{l}_l \prec \hat{l}_j = \hat{l}^* \prec z^* \prec \hat{h}^* = \hat{h}_l \prec \hat{h}_j \prec \hat{h}_i. \tag{37}$$

Since the above order also implies $\hat{l}_\ell \prec z^* \prec \hat{h}_\ell$ for $\forall \ell \in I_{S_3}$, then $z^*, \mathcal{R}^* \in K(C_\ell, y_0)$ for $\forall \ell \in I_{S_3}$. Thus, from (35), $z^*, \mathcal{R}^* \in K_3(S_3, y_0) = K^* = \text{cone}(\hat{l}^*, \hat{h}^*)$, provided $\hat{l}^*$ and $\hat{h}^*$ are chosen subject to (36). So far, it was assumed that $\hat{l}_i \prec \hat{h}_j$. If the unit vectors in $\Omega(S_3, y_0)$ and $\Lambda(S_3, y_0)$ are such that $\hat{l}_i \succ \hat{h}_j$, then there are no vectors that can satisfy the order $\hat{l}_i = \hat{l}^* \prec z^* \prec \hat{h}^* = \hat{h}_j$, and $K_3(S_3, y_0) = K^* = \emptyset$. □

APPENDIX D

LINEAR OPERATIONS FOR ORDERING UNIT VECTORS

This appendix illustrates a methodology for efficiently ordering sets of unit vectors according to a fixed frame of reference. Consider a set of unit vectors $\{\hat{u}_1, \ldots, \hat{u}_n\}$ with index set I. Any unit vector can be written in terms of its direction sine and cosine, namely, $\hat{u}_i = [\cos\gamma_i \ \sin\gamma_i]^T$, for $\forall i \in I$. We seek to order the unit vectors according to the xy-frame, therefore γ_i can also be viewed as the angle that $\hat{u}_i$ makes with the x-axis. Then, for any two unit vectors $\hat{u}_i$ and $\hat{u}_j$ in the first and fourth quadrant, $\hat{u}_i \prec \hat{u}_j$ if and only if $\sin\gamma_i < \sin\gamma_j$. From Proposition 4.2, it is of interest to obtain the first or last element of a list comprised of these unit vectors in ascending order: $\{\hat{u}_\jmath, \hat{u}_l, \ldots, \hat{u}_\imath\}$ with $\hat{u}_\jmath \preceq \hat{u}_l \preceq \ldots \preceq \hat{u}_\imath$. The first and last elements, $\hat{u}_\jmath$ and $\hat{u}_\imath$, can be obtained without ordering the entire set, using the following pairwise linear operations on the direction sines of the unit vectors:

$$\sin\gamma_\jmath = \frac{1}{2}\left[\sin\gamma_i + \sin\gamma_j - |\sin\gamma_i - \sin\gamma_j|\right], \tag{38}$$

$$\sin\gamma_\imath = \frac{1}{2}\left[\sin\gamma_i + \sin\gamma_j + |\sin\gamma_i - \sin\gamma_j|\right], \tag{39}$$

where $i \neq j$, $\forall i, \ j \in I$. It can be easily shown that the unit vectors generating cones with origin y_0 always lie in the first or fourth quadrant. Thus, the k-coverage cone $K_k(S_k, y_0)$ can be obtained by applying (38) to $\Omega(S_k, y_0)$ and by applying (39) to $\Lambda(S_k, y_0)$, as shown in Proposition 4.2.

The k-coverage cones defined with respect to the x, y', and x' axes (Section 4.3) can also be obtained by applying (38) and (39) to the corresponding sets of unit vectors, provided they first undergo a constant rotation. Let the rotation matrices

$$Q^{90} \equiv \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},\ Q^{180} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \text{and} \quad Q^{270} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{40}$$

denote clockwise rotations by 90 degree, 180 degree, and 270 degree, respectively. Then, (38) and (39) are applied to the rotated unit-vector sets, Ω^R and Λ^R, obtained by the following linear operations:

$$\Omega^R(S_k, x_0) \equiv \{\hat{h}_i^R \mid \hat{h}_i^R = Q^{90}\hat{h}_i,\ \forall \hat{h}_i \in \Omega(S_k, x_0)\}, \tag{41}$$

$$\Omega^R(S_k, y_0') \equiv \{\hat{h}_i^R \mid \hat{h}_i^R = Q^{180}\hat{h}_i,\ \forall \hat{h}_i \in \Omega(S_k, y_0')\}, \tag{42}$$

$$\Omega^R(S_k, x_0') \equiv \{\hat{h}_i^R \mid \hat{h}_i^R = Q^{270}\hat{h}_i,\ \forall \hat{h}_i \in \Omega(S_k, x_0')\}. \tag{43}$$

The sets $\Lambda^R(S_k, \cdot)$ are defined by substituting Ω with Λ in the above three equations. The rotated unit vector sets are only used to determine the indices $\jmath, \imath \in I_{S_k}$ of the unit vectors generating a k-coverage cone (Proposition 4.2). Once the indices are determined, unit vectors $\hat{h}_\jmath$ and $\hat{l}_\imath$ generate the cone $K_k(S_k, \cdot)$ (Section 5).

Appendix E
Proof of Theorem 5.1

We seek a measure μ on the set of tracks $\mathcal{K}_k(S, y_0)$ given by (9). Since $\mathcal{K}_k(S, y_0)$ is the union of m sets that may or may not be disjoint, we apply the principle of inclusion-exclusion [36]

$$\begin{aligned}\mu(\mathcal{K}_k(S, y_0)) &= \mu\left(\bigcup_{j=1}^{m} K_k(S_k^j, y_0)\right) \\ &= \sum_{j=1}^{m}(-1)^{j+1} \sum_{1 \le i_1 < \ldots < i_j \le m} \mu(K_k(S_k^{i_1}, y_0) \cap \ldots \cap K_k(S_k^{i_j}, y_0)),\end{aligned} \tag{44}$$

where

$$m = \binom{n}{k} = \frac{n!}{(n-k)!\ k!} \quad \text{and} \quad \sum_{1 \le i_1 < \ldots < i_j \le m}$$

is a sum over all the $[m!/(m-j)!\ j!]$ distinct integer j-tuples $(i_1, \ldots, i_j)$ satisfying $1 \le i_1 < \ldots < i_j \le m$. Also, $\mu(\cdot)$ denotes a measure on the set. Since the right-hand side of (44) is an intersection of cones, it is also a cone on which we can impose the Lebesgue measure μ.

Now, consider the intersection of cones $K_k(S_k^{i_1}, y_0) \cap \ldots \cap K_k(S_k^{i_j}, y_0)$ inside the inner summation in (44). $S_k^{i_l}$ denotes the i_lth k-subset of S, i_l is a positive integer between 1 and $i_j \le m$, and m is the total number of k-subsets in S. By the properties of cones, this intersection is also a cone and represents the set of tracks through y_0 that intersect all sensors in $S_p = \{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$. Based on the properties of k-subsets, this set must contain $k \le p \le n$ elements of S and, thus, is a p-subset of S. Based on the properties of k-coverage cones (Proposition 4.2), the set of line transversals of S_p through y_0 can be represented by the p-coverage cone $K_p(S_p, y_0) = K_p(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, y_0)$. Therefore, (44) can be written as

$$\begin{aligned}&\mu(\mathcal{K}_k(S, y_0)) = \\ &\sum_{j=1}^{m}(-1)^{j+1} \sum_{1 \le i_1 < \ldots < i_j \le m} \mu(K_p(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, y_0)),\end{aligned} \tag{45}$$

where p is the number of elements in the union of j k-subsets of S. Finally, since a Lebesgue measure on a k-coverage cone is its opening angle, a Lebesgue measure on $\mathcal{K}_k(S, y_0)$ is

$$\begin{aligned}\mathcal{T}_{y_0}^k &= \mu(\mathcal{K}_k(S, y_0)) \\ &= \sum_{j=1}^{m}(-1)^{j+1} \sum_{1 \le i_1 < \ldots < i_j \le m} \psi(S_k^{i_1} \cup \ldots \cup S_k^{i_j}, y_0).\end{aligned} \tag{46}$$

The opening angles in the above summation are given by (5) and (6) and, thus, $\mathcal{T}_{y_0}^k$ is a function of the sensors positions $X_S = \{s_1, \ldots, s_n\}$. □

Appendix F
Explicit Opening Angle Equations

Let $\psi = \psi(S_k, y_0)$ denote the opening angle of the k-coverage cone $K_k(S_k, y_0)$ for $\forall k$, $1 \le k \le n$, and $y_0 \equiv [0 \quad b_y]^T$. Then, according to Section 4.2, the cone is finitely generated by two unit vectors $\hat{l}_\imath$ and $\hat{h}_\jmath$ obtained from Λ and Ω such that $\imath, \jmath \in I_{S_k}$ and $\hat{l}_\imath \succeq \hat{l}_i \in \Lambda(S_k, y_0)$ and $\hat{h}_\jmath \preceq \hat{h}_i \in \Omega(S_k, y_0)$ for $\forall i \in I_{S_k}$ (as shown in Appendix D). Letting $\imath$ and $\jmath$ denote the indices of these unit vectors and using (16), the opening angle can be written explicitly as a function of X_S:

$$\left\{\begin{array}{l} \psi = H[\det(M_{\imath\jmath})] \cdot \sin^{-1}[\det(M_{\imath\jmath})], \\ \det(M_{\imath\jmath}) = \frac{1}{w_i^2 w_j^2}\{[x_i q_\imath + (y_i - b_y) r_\imath][x_\jmath r_\jmath + (y_\jmath - b_y) q_\jmath] \\ -[x_\jmath q_\jmath - (y_\jmath - b_y) r_\jmath][(y_i - b_y) q_\imath - x_i r_\imath]\}, \\ w_i \equiv \|v_i(y_0)\| = \sqrt{x_i^2 + (y_i - b_y)^2}, \quad q_i \equiv \sqrt{w_i^2 - r_i^2}, \\ i = \imath, \jmath, \end{array}\right\}, \tag{47}$$

where $X_S = \{s_i | i \in I_S\}$, $I_{S_k} \subset I_S$, and $s_i \equiv [x_i \quad y_i]^T$ for $\forall i$. The opening angles of the k-coverage cones defined with respect to the other axes are similarly derived by redefining the relative position vector v_i, as shown in [46].

Appendix G
Total Track Coverage

Consider the union of k-coverage cones $\mathcal{K}_k(S, y_0)$, representing the set of tracks through y_0 that are detected by at least k sensors in S and given by (9). Since all cones in this union are generated by objects in $\mathbb{R}_+^2$, $\mathcal{K}_k(S, y_0)$ only contains cones in the first and fourth quadrant of the xy-reference frame and its measure $\mathcal{T}_{y_0}^k$ in (18) is bounded from above by π. This upper bound corresponds to the case in which $\mathcal{K}_k(S, y_0)$ is a nonconvex cone and is a half space with $x \ge 0$. By induction, the measures $\mathcal{T}_{x_0}^k$, $\mathcal{T}_{y_0'}^k$, and $\mathcal{T}_{x_0'}^k$ are all bounded from above by π, for any value of the intercept. Thus, the upper bound on the track coverage function is obtained by substituting $\mathcal{T}_{y_0^\ell}^k = \mathcal{T}_{x_0^\ell}^k = \mathcal{T}_{y_0'^\ell}^k = \mathcal{T}_{x_0'^\ell}^k = \pi$ for any value of ℓ in (22):

$$\mathcal{T}_{\mathcal{A}}^{max} = \frac{1}{2}\sum_{\ell=1}^{N_2}(\pi+\pi) + \frac{1}{2}\sum_{\ell=0}^{N_1-1}(\pi+\pi) = N_2\pi + N_1\pi = \left(\frac{L_1+L_2}{\delta b}\right)\pi. \quad (48)$$

APPENDIX H
PROBABILITY OF TRACK DETECTION

Let the ray $\mathcal{R}_\alpha(b_y^\ell)$ denote a track with intercept value $b_y^\ell \in \mathcal{I}_y \equiv [0, L_2]$ and let D_k denote a cooperative detection event such that $D_k = 1$ if a target is detected by at least k sensors and $D_k = 0$ otherwise. Also, let $\Pr(b_y^\ell)$ denote the prior probability that a target enters $\mathcal{A}$ at $y_0^\ell = [0 \quad b_y^\ell]^T$. Then, the probability that a target enters $\mathcal{A}$ at y_0^ℓ, and is detected by k sensors is

$$\Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} = \Pr(b_y) \cdot \Pr(\mathcal{K}_k(S, y_0^\ell)), \quad (49)$$

where $\Pr(\mathcal{K}_k(S, y_0^\ell))$ denotes the probability that the track lies inside the set $\mathcal{K}_k(S, y_0^\ell)$. Now, assuming that all y-intercepts are equally likely, $\Pr(b_y^\ell) = \delta b/(L_2 + \delta b)$. Also, assuming that all directions $\alpha \in (-\pi/2, +\pi/2)$ are equally likely, (49) can be written as

$$\Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} = \frac{\delta b}{(L_2+\delta b)} \cdot \frac{1}{\pi}\sum_{j=1}^{m}(-1)^{j+1} \sum_{1\le i_1<\ldots<i_j\le m} \psi(S_p^{i_{1,j}}, y_0^\ell), \quad (50)$$

where $\psi(S_p^{i_{1,j}}, y_0^\ell)$ is the opening angle of the p-coverage cone of the p-subset of S that is defined as the union $\{S_k^{i_1} \cup \ldots \cup S_k^{i_j}\}$, for every tuple $(i_1, \ldots, i_j)$ in the inner summation (as shown in Appendix E).

Since sets $\mathcal{K}_k(S, y_0^\ell)$ with different values of y_0^ℓ are always disjoint, the probability that a target enters $\mathcal{A}$ through $\mathcal{I}_y$ and is detected by k sensors is

$$\Pr\{\mathcal{R} \cap \mathcal{I}_y \neq \emptyset, D_k = 1\} = \sum_{\ell=0}^{N_2}\Pr\{\mathcal{R}_\alpha(b_y^\ell), D_k = 1\} = \frac{\delta b}{\pi(L_2+\delta b)} \cdot \sum_{\ell=0}^{N_2}\sum_{j=1}^{m}(-1)^{j+1} \sum_{1\le i_1<\ldots<i_j\le m} \psi(S_p^{i_{1,j}}, y_0^\ell). \quad (51)$$

Similarly, the probability that $D_k = 1$ and the target intersects the sides $\mathcal{I}_x$, $\mathcal{I}_{y'}$, and $\mathcal{I}_{x'}$ can be obtained in terms of the opening angles. Then, the probability that a target traverses $\mathcal{A}$ and $D_k = 1$ is obtained by considering the probability of the union of intersecting sets [47]. The set of tracks that traverse $\mathcal{A}$ and are detected by at least k sensors is given by the union $\mathcal{K}_k(S, \mathcal{A})$ in (12). Since every track in this union intersects two sides of $\mathcal{A}$ and belongs to two k-coverage cones, the intersection of these cones is equal to its complement $P_{\mathcal{A}}^k(X_S) \equiv \Pr\{\mathcal{R} \cap \mathcal{A} \neq \emptyset, D_k = 1\}$ and, thus, (28) follows. It can be easily shown by substituting the same upper bounds used in Appendix F in (28) that, when the sensor network provides total track coverage, $P_{\mathcal{A}}^k = 1$.

ACKNOWLEDGMENTS

This work was supported by the US Office of Naval Research Young Investigator Program (Code 321).

REFERENCES

[1] Y. Zou and K. Chakrabarty, "A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks," *IEEE Trans. Computers,* vol. 54, no. 8, pp. 978-991, Aug. 2005.

[2] L.M. Kaplan, "Global Node Selection for Localization in a Distributed Sensor Network," *IEEE Trans. Aerospace and Electronic Systems,* vol. 42, no. 1, pp. 113-135, 2006.

[3] S. Iyengar and R. Brooks, "Special Issue Introduction—The Road Map for Distributed Sensor Networks in the Context of Computing and Communication," *J. Parallel and Distributed Computing,* vol. 64, no. 7, pp. 785-787, 2004.

[4] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Trans. Computers,* vol. 51, no. 12, pp. 1448-1453, Dec. 2002.

[5] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in Wireless Ad-Hoc Sensor Networks," *IEEE Trans. Computers,* vol. 52, no. 6, pp. 753-763, June 2003.

[6] S. Megerian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Worst and Best-Case Coverage in Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 4, no. 2, pp. 84-92, Jan./Feb. 2005.

[7] J. Ai and A.A. Abouzeid, "Coverage by Directional Sensors in Randomly Deployed Wireless Sensor Networks," *J. Combinatorial Optimization,* vol. 11, pp. 21-41, 2006.

[8] V. Isler, S. Khanna, and K. Daniilidis, "Sampling Based Sensor-Network Deployment," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems),* vol. 100, pp. 1780-1785, 2004.

[9] L. Benyuan and D. Towsley, "A Study of the Coverage of Large-Scale Sensor Networks," *Proc. First IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems,* pp. 475-483, 2004.

[10] N. Heo and P.K. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks," *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans,* vol. 35, no. 1, pp. 78-92, Jan. 2005.

[11] W. Huang, Y. Li, and C. Li, "Greedy Algorithms for Packing Unequal Circles into a Rectangular Container," *J. Operational Research Soc.,* vol. 56, pp. 539-548, 2005.

[12] J.-A. George, J.-M. George, and B. Lamer, "Packing Different-Sized Circles into a Rectangular Container," *European J. Operational Research,* vol. 84, pp. 693-712, 1995.

[13] S. Martinez and F. Bullo, "Optimal Sensor Placement and Motion Coordination for Target Tracking," *Automatica,* vol. 42, pp. 661-668, 2006.

[14] M. Marengoni, B.A. Draper, A. Hanson, and R.A. Sitaraman, "System to Place Observers on a Polyhedral Terrain in Polynomial Time," *Image and Vision Computing,* vol. 18, pp. 773-780, 1996.

[15] J. O'Rourke, *Art Gallery Theorems and Algorithms.* Oxford Univ. Press, 1987.

[16] T.A. Wettergren, "Statistical Analysis of Detection Performance for Large Distributed Sensor Systems," Technical Report ADA417136, Naval Undersea Warfare Center, http://stinet.dtic.mil, 2008.

[17] T.A. Wettergren, R.L. Streit, and J.R. Short, "Tracking with Distributed Sets of Proximity Sensors Using Geometric Invariants," *IEEE Trans. Aerospace and Electronic Systems,* vol. 40, no. 4, pp. 1366-1374, Oct. 2004.

[18] J.-P. LeCadre and G. Souris, "Searching Tracks," *IEEE Trans. Aerospace and Electronic Systems,* vol. 36, no. 4, pp. 1149-1166, 2000.

[19] B. Koopman, *Search and Screening: General Principles with Historical Applications.* Pergamon Press, 1980.

[20] H. Cox, "Cumulative Detection Probabilities for a Randomly Moving Source in a Sparse Field of Sensors," *Proc. 23rd Asilomar Conf. Signals, Systems and Computers,* pp. 384-389, 1989.

[21] J. Goodman, R. Pollack, and R. Wenger, "Geometric Transversal Theory," *New Trends in Discrete and Computational Geometry,* J. Pach, ed., pp. 163-198, Springer Verlag, 1991.

[22] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks," *Mobile Computing and Networking,* pp. 139-150, 2001.

[23] Q. Huang, "Solving an Open Sensor-Exposure Problem Using Variational Calculus," Technical Report WUCS-03-1, Washington Univ., http://www.cs.wustl.edu/qingfeng/papers/ExposureTRShort.pdf, 2003.
[24] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. Saluja, "Sensor Deployment for Detection of Targets Traversing a Region," *Mobile Networks and Applications,* vol. 8, pp. 453-461, Aug. 2003.
[25] E. Helly, "Über Mengen Konvexer Körper mit Gemeinschaftlichen Punkten," *Jahresbericht der Deutschen MathematikerVereinigung,* vol. 32, pp. 175-176, 1923.
[26] N. Megiddo, "Linear Programming in Linear Time When the Dimension Is Fixed," *J. ACM,* vol. 31, no. 1, pp. 114-127, 1984.
[27] E. Edelsbrunner, H. Maurer, F. Preparata, A. Rosenberg, E. Welzl, and D. Wood, "Stabbing Line Segments," *BIT Numerical Math.,* vol. 22, pp. 274-281, 1982.
[28] M. Atallah and C. Bajaj, "Efficient Algorithms for Common Transversals," *Information Processing Letters,* vol. 25, pp. 87-91, 1987.
[29] E. Edelsbrunner, "Finding Transversals for Sets of Simple Geometric Figures," *Theoretical Computer Science,* vol. 35, pp. 55-69, 1985.
[30] N. Ansari, J.-G. Chen, and Y.-Z. Zhang, "Adaptive Decision Fusion for Unequiprobable Sources," *IEEE Proc. Radar, Sonar, and Navigation,* vol. 144, no. 3, pp. 105-111, June 1997.
[31] D. Avis, J. Robert, and R. Wenger, "Lower Bounds for Line Stabbing," *Information Processing Letters,* vol. 33, pp. 59-62, 1989.
[32] D.P. Bertsekas, *Convex Analysis and Optimization.* Athena Scientific, 2003.
[33] H.F. Davis and A.D. Snider, *Vector Analysis.* William C. Brown, 1987.
[34] S. Skiena, "Generating *k*-Subsets," *Implementing Discrete Math.: Combinatorics and Graph Theory with Mathematica,* pp. 44-46, Addison-Wesley, 1990.
[35] S. Ferrari, "Track Coverage in Sensor Networks," *Proc. Am. Control Conf.,* pp. 2053-2059, 2006.
[36] S. Ross, *Introduction to Stochastic Dynamic Programming.* Academic Press, 1983.
[37] H.T. Croft, K.J. Falconer, and R.K. Guy, *Unsolved Problems in Geometry.* Springer-Verlag, 1991.
[38] D.P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 2007.
[39] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms.* Wiley Interscience, 2006.
[40] R.J. Vanderbei, "LOQO: An Interior Point Code for Quadratic Programming," *Optimization Methods and Software,* vol. 11, nos. 1-4, pp. 451-484, 1999.
[41] Mathworks, *Matlab Optimization Toolbox,* http://www.mathworks.com, function: fmincon, 2004.
[42] D. Hammond, "Sonobuoy Field Drift Prediction," "Naval Air Warfare Center Aircraft Division," Technical Report A115034, Patuxent River, Md., http://www.stormingmedia.us/11/1150/A115034.html, 2008.
[43] G.J. Juselis, "Station Keeping Buoy System," Secretary of the Navy Patent number: 5577942, http://www.google.com/patents?id=tMwhAAAAEBAJ, 1996.
[44] G. Petryk and M. Buehler, "Dynamic Object Localization via a Proximity Sensor Network," *Proc. IEEE/SICE/RSJ Int'l Conf. Multisensor Fusion and Integration for Intelligent Systems,* pp. 337-341, 1996.
[45] W. Ridely, "Automation of Buoy Positioning," *OCEANS,* vol. 17, pp. 179-183, 1985.
[46] K.A.C. Baumgartner, "Control and Optimization of Track Coverage in Underwater Sensor Networks," PhD thesis, Duke Univ., Dec. 2007.
[47] R.V. Hogg, J.W. McKean, and A.T. Craig, *Introduction to Mathematical Statistics.* Prentice Hall, 2005.

Kelli Baumgartner received the BS degree from Embry-Riddle Aeronautical University and the MS and PhD degrees from Duke University. She is an aerospace engineer in launch vehicle flight design and control at Analex Corp., a subsidiary of QinetiQ North America. While at Duke University, her principal research interests included distributed sensor networks and intelligent systems using Bayesian networks. She is a member of the IEEE, the AIAA, and the ASME.

Silvia Ferrari received the BS degree from Embry-Riddle Aeronautical University and the MA and PhD degrees from Princeton University. She is an assistant professor of mechanical engineering and materials science at Duke University, where she directs the Laboratory for Intelligent Systems and Controls (LISC). Her principal research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks. She is the recipient of the US Office of Naval Research Young Investigator award (2004), the US National Science Foundation CAREER award (2005), and the Presidential Early Career Award for Scientists and Engineers (PECASE) award (2006). She is a member of the IEEE, the ASME, the SPIE, and the AIAA.