A distributed deterministic annealing algorithm for limited-range sensor coverage

Andrew Kwok and Sonia Martínez

Abstract

This paper presents a distributed coverage algorithm for a network of mobile agents. Unlike previous work that uses a simple gradient descent algorithm, here we employ an existing deterministic annealing (DA) technique to achieve more optimal convergence values. We replicate the results of the classical DA algorithm while imposing a limited-range constraint to sensors. As the temperature is decreased, phase changes lead to a regrouping of agents, which is decided through a distributed task allocation algorithm. While simple gradient descent algorithms are heavily dependent on initial conditions, annealing techniques are generally less prone to this phenomena. The results of our simulations confirm this fact, as we show in the manuscript.

I. Introduction

The ability to autonomously deploy over a spatial region, as well as to dynamically adjust to single-point failures, gives mobile networks an advantage over static ones. This prompts the study of designing effective motion coordination algorithms for their unsupervised control [1]. A key area of interest regarding mobile sensor networks is deployment to maximize coverage [2], [3], [4], [5], [6].

However, most current methods for deployment, i.e. [2], [3], rely on gradient techniques to converge to an extremum of a cost function. As a result, the resulting final value of the cost function may not be the globally optimal one. Many annealing techniques exist to find a better optimal value of a cost function. Of these techniques, there are simulated annealing (SA) algorithms [7], as well as a more recent development, deterministic annealing (DA) [8].

A. Kwok and S. Martínez are with the department of Mechanical and Aerospace Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA, 92093. {ankwok, soniamd}@ucsd.edu

Unfortunately, these are centralized algorithms requiring global knowledge of the total state of the system.

Annealing algorithms differ from standard gradient algorithms through the addition of a temperature state. The goal, as in physical annealing, is to gradually lower this temperature, so that the internal configuration of the system is always at or near the lowest energy state. The SA and DA techniques feature *phase changes* as the temperatures are lowered past certain critical values, and we quantify these transitions for the distributed algorithm version.

A closely related work is that of Sharma et. al. [9]. The resulting algorithm discards information of other agents and resources that are far from a given agent. However, the algorithm still requires knowledge of all agents involved in the optimization to determine the information to discard.

In [10], SA was used to solve the clustering and formation control problems. That work also considered limited-range interactions, however, punctual long-range communication between agents was required. A cell decomposition of the environment had to be done a priori.

In this paper, we extend the DA algorithm of [8]. Here, we take that discrete DA algorithm to make it continuous in both space and time as well as spatially distributed. We strictly enforce that an individual agent can only sense the presence of other agents within a fixed radius. To do so, we introduce a spatial partition of the environment, and use this to develop a distributed local check of phase changes. Additionally, we introduce a task assignment algorithm to reassign vehicles according to phase changes. With the limited-range constraint, we achieve very similar results as in [8], [9]. Additionally, as this sensing radius increases, the algorithm recovers the original DA algorithm.

The paper is organized as follows. In section II, we introduce the limited range coverage problem, as well as provide an overview of the DA algorithm. In section III we derive the gradient direction for a limited-range DA algorithm, and continue in Section IV to provide a sufficient condition to distributively check for phase changes. We merge the two results in Section V by describing an algorithm for a network of autonomous agents to implement that includes a task allocation subroutine. We provide a simulation in Section VI as a proof of concept, followed by some concluding remarks.

II. NOTATION AND THE DA ALGORITHM

Let Q be a convex polytope in $\mathbb{R}^d$ including its interior, and let $\|\cdot\|$ denote the Euclidean norm. We will use $\mathbb{R}_{\geq 0}$ to denote the set of positive real numbers. A map $\phi : Q \to \mathbb{R}_{\geq 0}$, or a *distribution density function*, will represent a measure of a priori known information that some event takes place over Q. Equivalently, we consider Q to be the bounded support of the function ϕ. We will also denote the boundary of a set S as ∂S. The cardinality of S is denoted as $|S|$.

The proposed limited-range distributed DA algorithm is based on formations of agents (with leaders at $p_1, \ldots, p_n$) that split during phase changes. The algorithm finishes with formations of N single vehicles at positions $p_1, \ldots, p_N$. All agents have a limited sensing radius R_i, and they can communicate with other agents that are $2 \max_i R_i$ away. Let $B_i = \{q \in Q \mid \|p_i - q\| < R_i\}$ be an open ball of radius R_i around p_i intersected with Q.

We now briefly describe the minimization process of the DA scheme as well as compare this with the method in [2]. In [8], the end goal is to minimize a *distortion function*,

$$D = \int_Q \phi(q) \sum_{i=1}^{n} \mathbb{P}(p_i|q) f_i(\|q - p_i\|) dq \,, \tag{1}$$

where $f_i \colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a general metric (typically $f_i(x) = x^2$) and $\mathbb{P}(p_i|q)$ is the probability of a point q being associated with an agent p_i. However, (1) is not directly minimized. The Shannon entropy function is introduced:

$$H = -\int_Q \phi(q) \sum_{i=1}^{n} \mathbb{P}(p_i|q) \log \mathbb{P}(p_i|q) \,, \tag{2}$$

and the DA algorithm is a discrete-time algorithm that involves the minimization of the Lagrangian $F = D - TH$, where T is the temperature of the system. As temperature decreases, minimizing F becomes more similar to minimizing D. The association probabilities $\mathbb{P}(p_i|q)$ are derived from $\mathbb{P}^*(p_i|q) = \operatorname{argmin}_{\mathbb{P}(p_i|q)} F$. Then, the resulting $\mathbb{P}^*(p_i|q)$ are substituted into F to yield $\hat{F}$, and the optimal agent locations are given by $p_i^* = \operatorname{argmin}_{p_i} \hat{F}$.

As temperature decreases, the system undergoes *phase changes*. A phase change occurs when an equilibrium position p_i^* is no longer attractive in the presence of more than one sufficiently close agent. Rose in [8] provides a necessary and sufficient condition to detect phase changes, and we will provide an analogous check in the limited-range case.

In contrast the objective in [2] was to minimize (1) with trivial association probabilities determined by a Voronoi partition of Q. That is, the probability of $q \in Q$ being associated to p_i is 1 iff q is in its generalized Voronoi region.

As in [2], we choose to analyze the distributed DA coverage problem via general metrics $f_i \colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ such that f_i is Lipschitz and non-decreasing. Let $0 = R_{i,0} < R_{i,1} < \cdots < R_{i,m_i} = R_i$ be a finite sequence of radii. We assume that each f_i is of the form

$$f_i(x) = \sum_{\alpha=1}^{m_i} f_{i,\alpha}(x) 1_{[R_{i,\alpha-1}, R_{i,\alpha})}\,, \tag{3}$$

such that each $f_{i,\alpha}$ is differentiable and non-decreasing over $[R_{i,\alpha-1}, R_{i,\alpha})$. Additionally, we have for all α, $f_{i,\alpha}(R_{i,\alpha}) = f_{i,\alpha+1}(R_{i,\alpha})$, which enforces continuity of f_i.

In what follows we will consider the limited-range heterogeneous analogues of the area-maximizing and centroidal sensing metrics found in [2]. The sensing functions are, respectively,

$$f_i^a(x) = \left[\left(\frac{x}{R_i}\right)^c - 1\right] 1_{[0,R_i)}(x)\,, \tag{4}$$

$$f_i^m(x) = \left[x^2 - R_i^2\right] 1_{[0,R_i)}(x)\,, \tag{5}$$

where $c > 2$.

III. Limited-range DA Lagrangian gradient

In order to obtain a continuous-time version of the DA algorithm adapted to our coverage problem, we compute the gradient of the Lagrangian F with sensing functions (3) in this section. To do so, we first start with a derivation of the association probabilities, and then introduce a partition of Q that takes advantage of the limited-range nature of agent sensors.

A. Limited-range association probabilities

Similar to the original DA algorithm, we consider each point $q \in Q$ to have some probability of being associated with an agent at p_i. The probabilities, $\mathbb{P}(p_i|q)$ $i \in \{1, \ldots, n\}$, satisfy the following constraint for all $q \in Q$:

$$\sum_{i=1}^{n} \mathbb{P}(p_i|q) = 1\,. \tag{6}$$

Lemma 1: The association probability distribution that minimizes $F = D - TH$ and satisfies (6) is the Gibbs distribution

$$\mathbb{P}(p_i|q) = \frac{\exp\left[-\frac{f_i(\|q-p_i\|)}{T}\right]}{Z(q)}\,, i \in \{1,\ldots,n\}\,, \tag{7}$$

where the normalizing factor is:

$$Z(q) = \sum_{i=1}^{n} \exp\left[-\frac{f_i(\|q-p_i\|)}{T}\right]\,. \tag{8}$$

Proof: Following the DA derivation, we seek to minimize $F = D - TH$ first with respect to $\mathbb{P}(p_i|q)$ subject to (6). We employ the conservation of mass formula found in [2] to compute derivatives. Starting with (1),

$$\begin{aligned}\frac{\partial D}{\partial \mathbb{P}(p_i|q)} &= \int_Q \phi(q) f_i(\|q-p_i\|)dq + \int_{\partial Q} \phi(\gamma) \sum_i \mathbb{P}(p_i|\gamma) f_i(\|\gamma - p_i\|) n^T(\gamma) \frac{\partial \gamma}{\partial \mathbb{P}(p_i|q)} d\gamma \quad (\partial\gamma/\partial\mathbb{P}(p_i|q) \to 0)\\ &= \int_Q \phi(q) f(\|q-p_i\|)dq\,.\end{aligned}$$

Performing the same differentiation on (2),

$$\frac{\partial H}{\partial \mathbb{P}(p_i|q)} = -\int_Q \phi(q)[\log \mathbb{P}(p_i|q) + 1]dq\,.$$

To solve the constrained minimization problem, we use the Lagrange multipliers technique. Let $G = \sum_{i=1}^{n} \mathbb{P}(p_i|q) - 1$. In this way, $\frac{\partial G}{\partial \mathbb{P}(p_i|q)} = 1$. We then solve for

$$\nabla F = \lambda \nabla G\,, \tag{9}$$

$$G = 0\,. \tag{10}$$

Let $A = \int_Q \phi(q)dq$, then $\lambda(\nabla G)_i = \int_Q \frac{1}{A}\phi(q)\lambda dq$. Starting with (9), we have

$$\int_Q \phi(q)\left[f_i(\|q-p_i\|) + T\log \mathbb{P}(p_i|q) + T - \frac{\lambda}{A}\right]dq = 0\,.$$

The above is true if for all $q \in Q$,

$$\begin{aligned}0 &= f_i(\|q-p_i\|) + T\log \mathbb{P}(p_i|q) + T - \frac{\lambda}{A}\\ \mathbb{P}(p_i|q) &= \exp\left[\frac{\lambda}{AT} - 1 - \frac{f_i(\|q-p_i\|)}{T}\right]\,.\end{aligned}$$

Substituting the above into (10) results in

$$\exp\left[\frac{\lambda}{AT}-1\right]=\frac{1}{\sum_{i=1}^{n}\exp\left[-\frac{f_i(\|q-p_i\|)}{T}\right]},$$

and we can extract the results (7) and (8). ■

Remark 2: The function $Z(q)$ is continuous since each f_i is Lipschitz. This observation proves to be important for simplifying the analysis in future sections.

We can take the result (7) and substitute it back into F:

$$\begin{aligned}\hat{F}&=\int_Q\phi(q)\left[\sum_i\mathbb{P}(p_i|q)f_i(\|q-p_i\|)+T\mathbb{P}(p_i|q)\left(-\frac{f_i(\|q-p_i\|)}{T}-\log Z(q)\right)\right]dq\\&=-T\int_Q\phi(q)\log Z(q)dq\,,\end{aligned}\tag{11}$$

where we use the fact that $\sum_{i=1}^{n}\mathbb{P}(p_i|q)=1$.

B. Limited-range partition

For further analysis, it is advantageous to partition Q such that $Z(q)$ is differentiable over each region of this partition. We start by assuming that each sensing function f_i has the form from (3). We can define the set

$$A_{i,\alpha}=\{q\in Q\parallel R_{i,\alpha-1}\le\|q-p_i\|<R_{i,\alpha}\}\,.\tag{12}$$

This is the annulus centered at p_i with inner radius $R_{i,\alpha-1}$ and outer radius $R_{i,\alpha}$.

There are $M=\sum_{i=1}^{n}m_i$ of these sets, so we can equivalently enumerate the $A_{i,\alpha}$ as A_i for $i\in\{1,\dots,M\}$. Additionally, let β be the set of binary sequences of length M, i.e.: each $b_k\in\beta$, $k\in\{1,\dots,2^M\}$ is a finite sequence of zeros and ones.

Proposition 3: Let $\{D_k\}$ be a collection of sets such that for each $b_k\in\beta$,

$$D_k=\bigcap_{i=1}^{M}\{A_i\,\text{if}\,b_{k,i}=1\,;A_i^C\,\text{if}\,b_{k,i}=0\}\,.\tag{13}$$

Then, $\{D_k\}$ forms a partition of Q and $Z(q)$ is continuously differentiable in each D_k.

Proof: We show that $\{D_k\}$ forms a partition by verifying that: (i) $\bigcup_k D_k=Q$, and (ii) $D_k\cap D_\ell=\emptyset$ for $k\neq\ell$.

For the first criterion, by definition of the sets A_i, for any $q\in Q$ it is true that $q\in A_{i^*}$ for some $i^*\in I^*\subseteq\{1,\dots,M\}$. Then consider the binary sequence b_k such that $b_{k,i}=1$ for each

$i \in I^*$. Then by definition of the regions (13), $q \in D_k$. Since q is arbitrary, every point $q \in Q$ lies in some D_k, and so $Q = \bigcup_k D_k$.

For the next criterion, take two different regions D_k and D_ℓ. They are formed from the intersections

$$D_k = \bigcap_{i=1}^{M} \{A_i \text{ if } b_{k,i} = 1\,;\, A_i^C \text{ if } b_{k,i} = 0\}\,, \qquad D_\ell = \bigcap_{i=1}^{M} \{A_i \text{ if } b_{\ell,i} = 1\,;\, A_i^C \text{ if } b_{\ell,i} = 0\}\,,$$

respectively. Because $k \neq \ell$, the sequences $b_k \neq b_\ell$. Thus, for some index i^*, $b_{k,i^*} \neq b_{\ell,i^*}$. Wlog, suppose $b_{k,i^*} = 1$ and $b_{\ell,i^*} = 0$. Then we have

$$\begin{aligned} D_k \cap D_\ell &= \left[\bigcap_{i\neq i^*} \{A_i \text{ if } b_{k,i} = 1\,;\, A_i^C \text{ if } b_{k,i} = 0\} \cap A_{i^*}\right] \cap \left[\bigcap_{i\neq i^*} \{A_i \text{ if } b_{\ell,i} = 1\,;\, A_i^C \text{ if } b_{\ell,i} = 0\} \cap A_{i^*}^C\right] \\ &= \left[\bigcap_{i\neq i^*} \{A_i \text{ if } b_{k,i} = 1\,;\, A_i^C \text{ if } b_{k,i} = 0\} \cap \{A_i \text{ if } b_{\ell,i} = 1\,;\, A_i^C \text{ if } b_{\ell,i} = 0\}\right] \cap \left(A_{i^*} \cap A_{i^*}^C\right) \\ &= \emptyset\,. \end{aligned}$$

We have verified both properties, therefore $\{D_k\}$ is a partition of Q.

Next we show that the function $Z(q)$ is continuously differentiable over each D_k. From the definition (8), and assumption of the form of f_i in (3), we can write

$$Z(q) = \sum_{i=1}^{n} \sum_{\alpha=1}^{m_i} \exp\left[-\frac{f_{i,\alpha}(\|q - p_i\|) 1_{[R_{i,\alpha-1}, R_{i,\alpha})}(\|q - p_i\|)}{T}\right]\,. \tag{14}$$

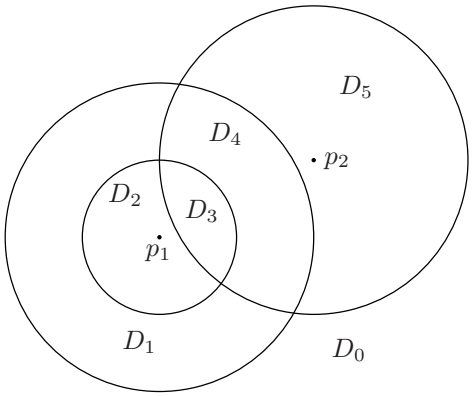
Additionally, each of the $f_{i,\alpha}$ are differentiable over the annulus centered at p_i with inner and outer radii of $R_{i,\alpha-1}$ and $R_{i,\alpha}$, respectively. Because the D_k are defined as the intersection of a subset of these annuli, $Z(q)$ is the sum of the same set of continuously differentiable $f_{i,\alpha}$ over each region D_k. Thus, $Z(q)$ is continuously differentiable over each D_k. ■

In the next section, we will use B_k to refer to the indices of the points p_i which form the region D_k. That is,

$$B_k = \{i \in \{1, \ldots, n\} \mid \|q - p_i\| < R_i,\, \forall\, q \in \mathrm{Int}(D_k)\}\,.$$

The regions D_k also have a convenient relation to each B_i.

Proposition 4: Let $A_{i,\alpha}$ be the annulus centered at p_i with inner radius $R_{i,\alpha-1}$ and outer radius $R_{i,\alpha}$. Each annulus $A_{i,\alpha}$ is *exactly* covered by a subcollection of $\{D_k\}$. We denote the indices of this subcollection as $C_{i,\alpha}$ such that $A_{i,\alpha} = \bigcup_{k\in C_{i,\alpha}} D_k$.



Variable	Description
$A_{1,2}$	$D_1 \cup D_4$
B_1	$\{1\}$
B_4	$\{1,2\}$
C_1	$\{1,2,3,4\}$
$C_{1,1}$	$\{2,3\}$
$C_{1,3}$	$\{0,5\}$
C_2	$\{3,4,5\}$
$C_{2,1}$	$\{3,4,5\}$

Fig. 1. Here we show an example for the notation we have introduced.

Proof: Since $\{D_k\}$ is a partition of Q, there exists a subcollection $\{D_k\}_{k\in I}$ such that $A_i \subseteq \bigcup_{k\in I} D_k$. Let I be the smallest index set such that this is true. Then by definition we have $D_k \cap A_i \neq \emptyset$ for each $k \in I$. Additionally, for each point p_i, the set of all annuli centered at p_i covers Q, see (12) and (3).

Now suppose there exists a D_k such that $A_i \cap D_k$ and $D_k \setminus A_i$ are both nonempty. Referencing the definition of D_k in (13), this D_k must be formed from the intersection of A_i and another annulus centered at p_i, but with different radii $R_{i,\alpha-1}, R_{i,\alpha}$. This intersection, however, is empty, and such a D_k cannot exist. ■

Corollary 5: Each ball B_i of radius R_i centered at p_i is *exactly* covered by a subcollection of $\{D_k\}$. We denote the set of indices corresponding to this subcollection as C_i such that $B_i = \bigcup_{k\in C_i} D_k$. •

We now introduce notation that will facilitate the derivation of the gradient direction and the critical temperature check. We have shown that a subset of $\{D_k\}$ forms a partition of each annulus $A_{i,\alpha}$ in Proposition 4. Thus, for a particular D_k, there may be portions of ∂D_k that are circular arcs centered at p_i with inner radius $R_{i,\alpha-1}$ and outer radius $R_{i,\alpha}$. We denote these circular arcs as $\mathrm{Arcs}(i,k)$.

C. Gradient formulation

The next step in the DA derivation is to optimize the Lagrangian $\hat{F}$ with respect to sensor positions p_i. Each agent in the network will use this result in order to compute its gradient direction.

Proposition 6: Given the Lagrangian (11), and sensing metrics of the form (3), the gradient of (11) is:

$$\frac{\partial \hat{F}}{\partial p_i} = -T \sum_{k \in C_i} \int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p_i} dq \,. \tag{15}$$

Proof: We begin by taking the following derivative (via the conservation of mass formula in [2]):

$$\frac{\partial \hat{F}}{\partial p_i} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p_i} dq + \int_{\partial D_k} \phi(\gamma_k) \log Z(\gamma_k) n^T(\gamma_k) \frac{\partial \gamma_k}{\partial p_i} d\gamma_k \right].$$

We now show the integrals over the boundaries ∂D_k vanish when summed over all k. Each γ_k that parametrizes the boundary of D_k is composed of circular arcs centered at various p_i. For a particular p_i, the derivative $\frac{\partial \gamma_k}{\partial p_i}$ is nonzero only when the γ_k parametrizes an arc centered at p_i. Since each $\mathrm{Arcs}(i,k)$ is a fixed radius from p_i,

$$\frac{\partial \gamma_k}{\partial p_i} = \begin{cases} I \,, & \gamma_k \in \mathrm{Arcs}(i,k) \,, \\ 0 \,, & \text{otherwise} \,. \end{cases}$$

Thus, only the integral along the boundaries $\mathrm{Arcs}(i,k)$ needs to be considered. The derivative is now simplified to

$$\frac{\partial \hat{F}}{\partial p_i} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p_i} dq + \int_{\mathrm{Arcs}(i,k)} \phi(\gamma_k) \log Z(\gamma_k) n^T(\gamma_k) d\gamma_k \right].$$

Since each arc in $\mathrm{Arcs}(i,k)$ is part of the boundary ∂D_k, Each arc in $\mathrm{Arcs}(i,k)$ is shared between two regions D_k and D_ℓ. Thus, there will be two integrals over each $\mathrm{Arcs}(i,k)$: one from D_k and one from D_ℓ. For these two integrals, the normal vector $n(\gamma_k)$ will be equal and opposite. Additionally the function $Z(q)$ is continuous over Q, so the sum of these two integrals will be zero. The derivative is simplified to

$$\frac{\partial \hat{F}}{\partial p_i} = -T \sum_k \int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p_i} dq \,.$$

We now show that the derivative $\frac{\partial Z}{\partial p_i}$ is zero if $q \notin B_i$. Recall from the limited-range assumption that each sensing function $f_{i,\alpha}(x)$ is a constant if $x \geq R_i$. Therefore Z as in (14) has no dependence on p_i if $\|q - p_i\| \geq R_i$. With this realization, we obtain the result (15). ■

Observe that the set V can be included as part of the integrals because it consists of isolated points.

Remark 7: For the area-maximizing case (4), we begin by computing the derivative $\frac{\partial Z}{\partial p_i}$.

$$\begin{aligned}\frac{\partial Z}{\partial p_i} &= \sum_{j=1}^{n} \frac{\partial}{\partial p_i} \exp\left[-\frac{f_j^a(\|q-p_j\|)}{T}\right] \\ &= -\frac{1}{T}\frac{\partial}{\partial p_i}[f_i^a(\|q-p_i\|)] \exp\left[-\frac{f_i^a(\|q-p_i\|)}{T}\right].\end{aligned}$$

As an aside, we have the following result:

$$\begin{aligned}\frac{\partial}{\partial p_i}[\|q-p_i\|^c] &= \frac{\partial}{\partial p_i}\left[\left((q-p_i)^T(q-p_i)\right)^{c/2}\right] \\ &= \frac{c}{2}\left(\|q-p_i\|^2\right)^{c/2-1}\frac{\partial}{\partial p_i}\left[\|q\|^2 - 2q^T p_i + \|p_i\|^2\right] = c\|q-p_i\|^{c-2}(p_i-q)^T.\end{aligned}$$

Then using the above result when taking the derivative of (4), we obtain the following gradient expression

$$\begin{aligned}\frac{\partial \hat{F}}{\partial p_i} &= -T\sum_{k\in C_i}\int_{D_k}\phi(q)\frac{c\|q-p_i\|^{c-2}}{R_i^c T}(q-p_i)^T\mathbb{P}(p_i|q)dq \\ &= -\frac{c}{R_i^c}\sum_{k\in C_i}\int_{D_k}\phi(q)\|q-p_i\|^{c-2}(q-p_i)^T\mathbb{P}(p_i|q)dq\,. \qquad (16)\end{aligned}$$

Here we see that the weight $\|q-p_i\|^{c-2}$ serves to amplify the density ϕ for points close to the boundary of B_i while neglecting the value of ϕ close to p_i for large c. This achieves the area-maximizing effect that we seek. •

Remark 8: We can compute the derivative $\frac{\partial Z}{\partial p_i}$ using the sensing function (5) in a similar manner. We then obtain the gradient (15) to be:

$$\frac{\partial \hat{F}}{\partial p_i} = -2\sum_{k\in C_i}\int_{D_k}\phi(q)(q-p_i)^T\mathbb{P}(p_i|q)dq\,. \qquad (17)$$

This is similar to the gradient expression for the mixed coverage case in [2], with the addition of the association probabilities $\mathbb{P}(p_i|q)$ as an extra weighting factor. •

D. Constant factor approximation

Thus far, we have developed a gradient direction for agents to follow in order to minimize (11), which for low temperatures, minimizes the underlying distortion function (1). We now relate how minimizing (11) for $\max_i R_i < \mathrm{diam}(Q)$ bears some relation to minimization of (11) for when $R_i > \mathrm{diam}(Q)$ for all i. In other words, we demonstrate the relation between the gradient-descent of the limited-range DA algorithm with the gradient descent of the original DA algorithm.

We compare the mixed case (5) as this is most similar to the distance metric found in [8],

$$F_0 = -T \int_Q \phi(q) \log Z_0(q) dq \,, \qquad Z_0(q) = \sum_i \exp\left[-\frac{\|q - p_i\|^2}{T}\right] . \tag{18}$$

This proposition makes it clear that minimization of (11) is equivalent to minimization of the analogous function in the original DA case as sensing radii increase.

Proposition 9: Let F_0 be defined as in (18). Additionally, let $\hat{F}$ be defined as from (11). Then, the following is true:

$$\hat{F} + \min_i R_i^2 \leq F_0 \leq \hat{F} + \mathrm{diam}^2(Q) \,. \tag{19}$$

Proof: Let $f_i(x) = x^2$ and let $f_i^m(x)$ be defined as in (5). Additionally, let $\alpha = \min_i R_i$ and $d = \mathrm{diam}(Q)$. Then it is true that $-f_i^m(x) - d^2 \leq -f_i(x) \leq -f_i^m(x) - \alpha^2$. Since the exponential and logarithmic functions are monotonic increasing, the following statements hold:

$$\sum_i \exp\left[-\frac{f_i^m(x) + d^2}{T}\right] \leq \sum_i \exp\left[-\frac{f_i(x)}{T}\right] \leq \sum_i \exp\left[-\frac{f_i^m(x) + \alpha^2}{T}\right] ,$$

$$-\frac{d^2}{T} + \log \sum_i \exp\left[-\frac{f_i^m(x)}{T}\right] \leq \log \sum_i \exp\left[-\frac{f_i(x)}{T}\right] \leq -\frac{\alpha}{T} + \log \sum_i \exp\left[-\frac{f_i^m(x)}{T}\right] .$$

Substitute each of these into the expression for F to obtain:

$$-T \int_Q \phi(q) \left[-\frac{\alpha^2}{T} + \log Z(q)\right] dq \leq -T \int_Q \phi(q) \log Z_0(q) dq \leq -T \int_Q \phi(q) \left[-\frac{d^2}{T} + \log Z(q)\right] dq \,.$$

The result (19) follows since ϕ can be normalized over Q. ■

IV. Limited-range DA phase changes

As temperature decreases, the equilibrium points of $\hat{F}$ under the evolution of (15) become unstable. When this happens a phase change occurs and we say that we have reached a critical temperature. We present a sufficient condition for agents to individually check if they have

reached a critical temperature value under both area-maximizing and mixed centroidal-area coverage.

Using a similar argument as in [8], we enlarge the group of leaders $\{p_1, \ldots, p_n\}$ with a set of virtual agents $\{p_{n+1}, \ldots, p_l\}$ so that for all $j \in \{n+1, \ldots, l\}$, $p_j = p_i$ for some $i \in \{1, \ldots, n\}$. Then we introduce perturbations $\Psi = (\psi_1, \ldots, \psi_l) \in \mathbb{R}^{2l}$. Given a scaling factor ϵ, consider the perturbed agent locations, $x_i = p_i + \epsilon\psi_i$, for $i \in \{1, \ldots, l\}$. Critical points of $\hat{F}$ correspond to configurations where $\frac{d\hat{F}(x_1,\ldots,x_l)}{d\epsilon}\Big|_{\epsilon=0} = 0$. However, those configurations fail to be a minimum when the second derivative $\frac{d^2\hat{F}}{d\epsilon^2}\Big|_{\epsilon=0} \leq 0$. We now find the second derivative. Consider the partition $\{D_k\}$ associated with the $\{x_i\}$, $i \in \{1, \ldots, l\}$. The first derivative of the Lagrangian (11) with respect to ϵ is

$$\frac{d\hat{F}}{d\epsilon} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial \epsilon} dq + \int_{\partial D_k} \phi(\gamma_k) \log Z(\gamma_k) n^T(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \right].$$

Using the same reasoning as before when computing the gradient (15), the integrals over the boundaries ∂D_k cancel when summed over all k. Taking another derivative with respect to ϵ,

$$\begin{aligned} \frac{d^2\hat{F}}{d\epsilon^2} = -T \sum_k \Bigg\{ & \int_{D_k} \phi(q) \left[-\frac{1}{Z^2(q)} \left(\frac{\partial Z}{\partial \epsilon} \right)^2 + \frac{1}{Z(q)} \frac{\partial^2 Z}{\partial \epsilon^2} \right] dq \\ & + \int_{\partial D_k} \phi(\gamma_k) \frac{1}{Z(\gamma_k)} \frac{\partial Z}{\partial \epsilon} n^T(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \Bigg\}. \end{aligned} \tag{20}$$

Unfortunately, the same convenient cancellation of the boundary terms may not occur here. Since each $f_i(x)$ is only Lipschitz, and $Z(q)$ is composed of a sum of exponential of f_i, $Z(q)$ is also only Lipschitz. The derivative of Z evaluated at one side of the boundary ∂D_k may not be the same as it is evaluated on the other side.

Let $y_i = q - x_i$ to reduce the amount of notation. The derivative $\frac{dZ}{d\epsilon}$ is computed as follows:

$$\frac{dZ}{d\epsilon} = \sum_{i=1}^{l} \sum_{\alpha=1}^{m_i} -\frac{1}{T} \frac{\partial f_{i,\alpha}}{\partial \epsilon} \exp\left[-\frac{f_{i,\alpha}(\|y_i\|) 1_{[R_{i,\alpha-1}, R_{i,\alpha})}(\|y_i\|)}{T} \right] 1_{[R_{i,\alpha-1}, R_{i,\alpha})}(\|y_i\|).$$

Again note that this derivative may not be continuous if $\frac{\partial f_{i,\alpha}}{\partial \epsilon}$ is not continuous. In a particular region D_k, the above simplifies to

$$\frac{dZ}{d\epsilon} = \sum_{j \in B_k} -\frac{1}{T} \frac{\partial f_{j,\alpha}}{\partial \epsilon} \exp\left[-\frac{f_{j,\alpha}(\|y_j\|)}{T} \right]. \tag{21}$$

This is because the indicator function evaluates to zero if $\|y_i\| = \|q - x_i\| \leq R_i$, and the index set B_k captures all such x_i that satisfy this condition. It should be noted that the α in $f_{i,\alpha}$ may

vary. This α indexes the sensing metric associated with an annulus centered at an agent x_j. Since the region D_k is fixed, the associated annulus index corresponding to the region D_k may change for various $j \in B_k$.

Continuing, the second derivative is:

$$\frac{d^2 Z}{d\epsilon^2} = \sum_{j \in B_k} \left\{ \left[\left(\frac{1}{T} \frac{\partial f_{j,\alpha}}{\partial \epsilon} \right)^2 - \frac{1}{T} \frac{\partial^2 f_{j,\alpha}}{\partial \epsilon^2} \right] \exp \left[-\frac{f_{j,\alpha}(\|y_j\|)}{T} \right] \right\}. \tag{22}$$

Since $y_i = q - p_i - \epsilon \psi_i$, using the chain rule, $\frac{\partial f_{i,\alpha}}{\partial \epsilon} = -\psi_i^T \frac{\partial f_{i,\alpha}}{\partial y_i}$ and $\frac{\partial^2 f_{i,\alpha}}{\partial \epsilon^2} = \psi_i^T \frac{\partial^2 f_{i,\alpha}}{\partial y_i^2} \psi_i$.

We substitute the results (21) and (22) into (20), and note that $\frac{1}{Z} \exp \left[-\frac{f_{i,\alpha}}{T} \right] = \mathbb{P}(x_i|q)$ to get:

$$\begin{aligned} \frac{d^2 \hat{F}}{d\epsilon^2} = -T \sum_k \Bigg\{ \int_{D_k} \phi(q) \Bigg(& -\frac{1}{T^2} \left(\sum_{j \in B_k} \frac{\partial f_{j,\alpha}}{\partial \epsilon} \mathbb{P}(x_j|q) \right)^2 \\ & + \sum_{j \in B_k} \left[\left(\frac{1}{T} \frac{\partial f_{j,\alpha}}{\partial \epsilon} \right)^2 - \frac{1}{T} \frac{\partial^2 f_{j,\alpha}}{\partial \epsilon^2} \right] \mathbb{P}(x_j|q) \Bigg) dq \\ & - \frac{1}{T} \int_{\partial D_k} \phi(\gamma_k) \sum_{j \in B_k} \left(\frac{\partial f_{j,\alpha}}{\partial \epsilon} \mathbb{P}(p_j|\gamma_k) \right) n^T(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \Bigg\}. \end{aligned} \tag{23}$$

The check for critical temperature is to numerically compute $\frac{d^2 \hat{F}}{d\epsilon^2}\Big|_{\epsilon=0}$ at an equilibrium configuration. The equilibrium configurations occurs when $\frac{\partial \hat{F}}{\partial p_i} = 0$ for all i, or equivalently, when $\frac{d\hat{F}}{d\epsilon}\Big|_{\epsilon=0} = 0$. If the second derivative is negative, then the equilibrium configuration is unstable, and that signifies that we are below a critical temperature value. To simplify the critical temperature check and make it spatially distributed, we consider the following perturbation. Let $S_i \subseteq \{1, \ldots, m\}$ be such that $j \in S_i$ implies $p_j = p_i$. We define Ψ_i to be

$$\Psi_i = \Big\{ (\psi_1, \ldots, \psi_m) \mid \psi_j = 0, \, \forall \, j \notin S_i; \sum_{j \in S_i} \psi_j = 0 \Big\}. \tag{24}$$

If the critical temperature has not yet been reached, then these coincident agents (i.e., leaders and virtual agents) will remain together. Otherwise, the coincident agents are at an unstable equilibrium point, and any perturbation will force them apart. By using this particular perturbation, we will obtain a sufficient condition for critical temperature. We will now take the above results and consider the metric functions (4) and (5).

A. Area metric

For the area-maximizing metric, we present the following check for critical temperature.

Proposition 10: Critical temperature for the area-maximizing DA algorithm has been reached if for $i \in \{1, \ldots, n\}$, any of the following matrices $\mathcal{F}_i$ are negative definite:

$$\mathcal{F}_i = \sum_{k \in C_i} \int_{D_k} \phi(q) \mathbb{P}(p_i|q) \left(\left[(c-2) - \frac{c\|q-p_i\|^c}{T R_i^c} \right] (q-p_i)(q-p_i)^T + \|q-p_i\|^2 I \right) dq \,, \quad (25)$$

and $\dot{p}_i = 0$ for all $i \in \{1, \ldots, n\}$.

Proof: With this choice of perturbations, consider the case using the area metric f_i^a from (4). We fix an i such that the perturbation (24) is true and compute $\frac{d^2 \hat{F}}{d\epsilon^2}\Big|_{\epsilon=0}$ using (23).

The derivatives $\frac{\partial f_i^a}{\partial y_i}$ and $\frac{\partial^2 f_i^a}{\partial y_i^2}$ are:

$$\begin{aligned}
\frac{\partial f_i^a}{\partial y_i} &= \frac{1}{R_i^c} \frac{c}{2} (\|y_i\|^2)^{c/2-1} (2y_i^T) = \frac{c}{R_i^c} \|y_i\|^{c-2} y_i^T \,, \\
\frac{\partial^2 f_i^a}{\partial y_i^2} &= \frac{c}{R_i^c} \left(\frac{c}{2} - 1 \right) (\|y_i\|^2)^{c/2-2} (2y_i) y_i^T + \frac{c}{R_i^c} \|y_i\|^{c-2} I \\
&= \frac{c}{R_i^c} \|y_i\|^{c-4} \left[(c-2) y_i y_i^T + \|y_i\|^2 I \right] \,.
\end{aligned}$$

Since $y_i = q - p_i - \epsilon\psi_i$, when $\epsilon = 0$, $y_j = q - p_j$ for all $j \in S_i$. However, since $p_j = p_i$, $y_j = q - p_i$ for all $j \in S_i$. Similarly, the association probabilities $\mathbb{P}(x_j|q) = \mathbb{P}(p_i|q)$ for all $j \in S_i$.

When considering the perturbations (24) on (23), we note the following simplification. In (23), there is a sum over $j \in B_k$ involving the first and second derivatives, $\frac{\partial f_{j,\alpha}}{\partial \epsilon} = \psi_j \frac{\partial f_{j,\alpha}}{\partial y_j}$ and $\frac{\partial^2 f_{j,\alpha}}{\partial \epsilon^2} = \psi_j^T \frac{\partial^2 f_{j,\alpha}}{\partial y_j^2} \psi_j$ respectively. However, if $j \notin S_i$ then $\psi_j = 0$. Therefore instead of summing over $j \in B_k$, we can equivalently sum over $j \in S_i$.

With the above results, the second derivative (23) evaluated at $\epsilon = 0$ can be simplified as follows:

$$\begin{aligned}
\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} &= -T\sum_{k\in C_i}\Bigg\{\int_{D_k}\phi(q)\Bigg(-\frac{1}{T^2}\left(-\frac{c\|q-p_i\|^{c-2}}{R_i^c}(q-p_i)^T\mathbb{P}(p_i|q)\cancelto{0}{\sum_{j\in S_i}\psi_j^T}\right)^2 \\
&\qquad + \sum_{j\in S_i}\left[\left(\frac{1}{T}\frac{\partial f_{j,\alpha}}{\partial\epsilon}\right)^2 - \frac{1}{T}\frac{\partial^2 f_{j,\alpha}}{\partial\epsilon^2}\right]\mathbb{P}(p_i|q)\Bigg)dq \\
&\qquad - \frac{1}{T}\int_{\partial D_k}\phi(\gamma_k)\left(-\frac{c\|\gamma_k-p_i\|^{c-2}}{R_i^c}(\gamma_k-p_i)^T\mathbb{P}(p_i|\gamma_k)\cancelto{0}{\sum_{j\in S_i}\psi_j^T}\right)n^T(\gamma_k)\frac{\partial\gamma_k}{\partial\epsilon}d\gamma_k\Bigg\} \\
&= -T\sum_{k\in C_i}\Bigg\{\int_{D_k}\phi(q)\left(\sum_{j\in S_i}\left[\left(\frac{1}{T}\frac{\partial f_{j,\alpha}}{\partial\epsilon}\right)^2 - \frac{1}{T}\frac{\partial^2 f_{j,\alpha}}{\partial\epsilon^2}\right]\mathbb{P}(p_i|q)\right)dq \\
&= -\sum_{j\in S_i}\sum_{k\in C_i}\int_{D_k}\phi(q)\mathbb{P}(p_i|q)\Bigg[\frac{1}{T}\left(-\frac{c\|q-p_i\|^{c-2}}{R_i^c}\psi_j^T(q-p_i)\right)^2 \\
&\qquad - \psi_j^T\frac{c}{R_i^c}\|q-p_i\|^{c-4}\left[(c-2)(q-p_i)(q-p_i)^T + \|q-p_i\|^2 I\right]\psi_j\Bigg]dq\,.
\end{aligned}$$

Factoring out ψ_j from the left and right sides and using the substitution (25), the second derivative evaluated at $\epsilon = 0$ is:

$$\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} = \frac{c\|q-p_i\|^{c-4}}{R_i^c}\sum_{j\in S_i}\psi_j^T\mathcal{F}_i\psi_j\,.$$

It is clear now that in order for an equilibrium configuration to be stable in the area-maximizing case, the matrix quantity in (25) must be positive definite. ■

B. Mixed metric

We perform the same analysis for the mixed centroidal-area coverage sensing metric (5). This metric is most similar to that found in [8] and [9].

Proposition 11: Critical temperature for the centroidal-area DA algorithm has been reached if for $i \in \{1,\ldots,n\}$, any of the following matrices $\mathcal{F}_i$ are negative definite:

$$\mathcal{F}_i = \sum_{k\in C_i}\int_{D_k}\phi(q)\mathbb{P}(p_i|q)\left[I - \frac{2}{T}(q-p_i)(q-p_i)^T\right]dq\,, \tag{26}$$

and $\dot{p}_i = 0$ for all $i \in \{1,\ldots,n\}$.

Proof: The derivatives $\frac{\partial f_i^m}{\partial y_i}$ and $\frac{\partial^2 f_i^m}{\partial y_i^2}$, when $\|y_i\| \leq R_i$, are:

$$\frac{\partial f_i^m}{\partial y_i} = 2y_i^T$$
$$\frac{\partial^2 f_i^a}{\partial y_i^2} = 2I\,.$$

Since $y_i = q - p_i - \epsilon\psi_i$, when $\epsilon = 0$, $y_j = q - p_i$ for all $j \in S_i$. Similarly, the association probabilities $\mathbb{P}(x_j|q) = \mathbb{P}(p_i|q)$ for all $j \in S_i$. Therefore, with the perturbations (24) and the mixed metric (5), the second derivative (23) evaluated at $\epsilon = 0$ can be simplified as follows:

$$\begin{aligned}
\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} &= -T\sum_{k\in C_i}\left\{\int_{D_k}\phi(q)\left(-\frac{1}{T^2}\left(-2(q-p_i)^T\mathbb{P}(p_i|q)\cancelto{0}{\sum_{j\in S_i}}\psi_j\right)^2\right.\right.\\
&\quad\left.+\sum_{j\in S_i}\left[\left(-\frac{2}{T}\psi_j^T(q-p_i)\right)^2-\frac{2}{T}\psi_j^T I\psi_j\right]\mathbb{P}(p_i|q)\right)dq\\
&\quad\left.-\frac{1}{T}\int_{\partial D_k}\phi(\gamma_k)\left(-2\mathbb{P}(p_i|\gamma_k)(\gamma_k-p_i)^T\cancelto{0}{\sum_{j\in S_i}}\psi_j\right)n^T(\gamma_k)\frac{\partial\gamma_k}{\partial\epsilon}d\gamma_k\right\}\\
&= -\sum_{k\in C_i}\left\{\int_{D_k}\phi(q)\mathbb{P}(p_i|q)\left[\frac{4}{T}\sum_{j\in S_i}\left(-\psi_j^T(q-p_i)\right)^2-2\psi_j^T I\psi_j\right]dq\right\}\\
&= 2\sum_{j\in S_i}\sum_{k\in C_i}\int_{D_k}\phi(q)\mathbb{P}(p_i|q)\left[\psi_j^T I\psi_j-\frac{2}{T}\left(\psi_j^T(q-p_i)\right)^2\right]dq\,.
\end{aligned}$$

Factoring out ψ_j from the left and right sides and using the substitution (26), the second derivative evaluated at $\epsilon = 0$ is:

$$\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} = 2\sum_{j\in S_i}\psi_j^T\mathcal{F}_i\psi_j\,.$$

It is clear now that in order for an equilibrium configuration to be stable in the area-maximizing case, the matrix quantity in (26) must be positive definite. ■

V. Distributed implementation

We have so far demonstrated how a network of agents can descend the gradient and check for phase changes in a distributed DA algorithm. However, we still must provide a distributed method for implementing these phase changes.

The DA algorithm begins with one active agent, and the other agents moving in formation with it. A formation will split in two if its critical temperature is reached. The agents following

Algorithm 1: Distributed DA algorithm for each agent

$T \leftarrow$ initial temperature

while $T > T_{min}$ **or** $n < N$ **do**

  while *floodMax(*$\|\dot{p}_i\|$*)* $> \epsilon$*)* **do**
    $\dot{p}_i \leftarrow -$computeGradient()

  end

  if *checkSplit()* $==$ *true* **then** flood("T_c reached")

  if *received "*T_c *reached"* **then**
    doTaskAssign() $N - n$ times

  end

  $T \leftarrow \alpha T$

end

doNormalCoverage()

in formation are divided evenly between the current formation leader and a new formation leader. After the first phase change, it is possible that future phase changes occur at an agent who is by itself. Therefore, this agent must communicate its desire for an additional companion, and the network of agents must distributively assign an inactive agent to this task. We propose a task-assignment algorithm to accomplish this.

We provide a possible scheme under the following assumptions: (1) Agents have knowledge of the total number of formations n and the total number of agents N, (2) The communication graph between all active agents is connected, (3) Each active agent knows the number of inactive agents traveling with it, and (4) All agents have knowledge of the initial temperature, and the cooling factor α.

Connectivity of the communication graph is important because both the temperature and the total number of active agents must be constant across all agents. We assume that if the graph is connected, the agents can agree on the current temperature, and determine through a flooding algorithm (see [11]) the number of active agents n at any point in time. Additionally, agents must wait for the flooding algorithms to terminate; the worst case is proportional to the diameter of the communication graph.

The scheme also uses primitives for flooding or agreement over the network to acquire global

Algorithm 2: Task assignment algorithm for each agent

$a_i \leftarrow$ number of agents in formation

if *checkSplit() == true* **then**

  if $a_i == 0$ **then**

    flood("need companion at p_i")

    $M \leftarrow$ positions p_j of replies for help

    if $m_i == null, \forall\, m_i \in M$ **then**

      return

    else

      $J \leftarrow$ sortAscending($\{\|p_i - p_j\|\}, j \in M$)

      $j^* \leftarrow$ removeFirst(J)

      sendMsg("request companion", j^*)

      flood("increment n by 1")

    end

  else

    split formation evenly

    flood("increment n by 1")

  end

else `// no splitting at` p_i

  $M \leftarrow$ received companion requests p_j

  $J \leftarrow$ sortAscending($\{\|p_i - p_j\|\}, j \in M$)

  if $a_i == 0$ **then**

    sendMsg(null, $\forall\, j \in J$)

  else

    while *length(J)* > 0 **and** $a_i > 0$ **do**

      $j^* \leftarrow$ removeFirst(J)

      sendMsg("help available from p_i", j^*)

      $a_i \leftarrow a_i - 1$

    end

  end

end

information. We define `flood(`msg`)` to be an algorithm that floods a message over the entire network, such that after its completion, each active agent will have knowledge of msg (possibly the `null` message). Messages to a particular agent i can be sent with `sendMsg(`msg`,` i`)`. We also define `floodMax(`x_i`)` as a flooding method to determine $\max_{i\in\{1,\dots,n\}} x_i$ over the entire network as in [11]. We let `computeGradient()` be the function that computes (15), and we let `checkSplit()` be the function that determines if a critical temperature has been reached as in (25) and (26). Finally, we introduce `doNormalCoverage()` to mean to perform limited-range coverage as from [2].

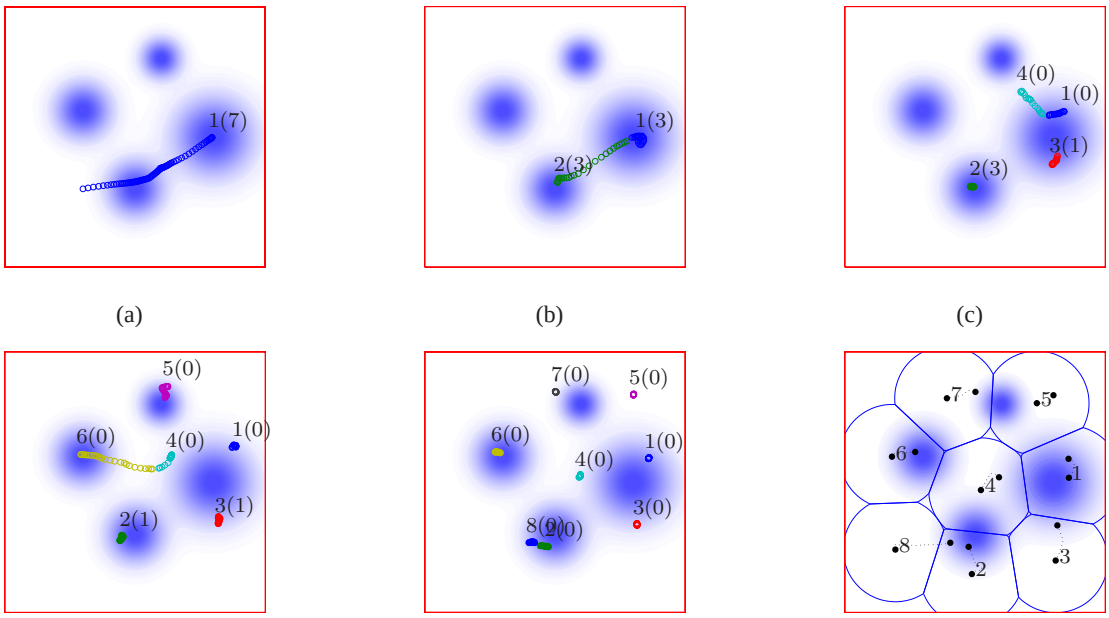
The distributed DA algorithm can informally be described as follows, see Algorithm 1. Starting with a single formation, and a high initial temperature, formations descend the gradient (15). When all agents agree they are stationary, they individually check for phase changes and, if necessary, implement Algorithm 2 $N - n$ times to guarantee the assignment of all companion requests. The temperature is lowered, regardless of whether or not there was a phase change, and the gradient descent is continued. This process repeats until the system temperature is below a minimum temperature threshold T_{min} or if $n = N$. Once this happens, the agents perform the normal coverage algorithm described in [2], as this is equivalent to having $T = 0$.

Algorithm 2 outlines the task assignment algorithm for agents who are in need of a companion to split. Roughly speaking, there are three rounds of communication where an agent broadcasts its need for a companion, other agents reply if they can help, and finally a handshake is formed with the agent transfer. In this algorithm, n is incremented for every new formation, and this command is flooded over the network. This algorithm has a finite termination time upper bounded by $3n + n(N - n)$ messages passed.

VI. Simulations

We present a simulation of the limited-range DA algorithm using the area-maximizing sensing function (4) in Figure 2. The total number of agents is $N = 8$ and the square region Q has length 10 per side. Each agent has a sensing radius of $R = 2$. The initial temperature is $T = 10$ and the annealing factor $\alpha = 0.9$ in this simulation.

As the temperature decreases, we see the agents split and continue to increase coverage area. Once the minimum temperature is reached, the coverage algorithm of [2] is conducted, and Figure 2 (f) shows the limited-range Voronoi partitions separating each agent.



(a) (b) (c)

(d) (e) (f)

Fig. 2. A typical run of the limited-range DA algorithm. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). The temperatures begin at $T = 10$ (a), then decrease: $T = 0.15$ (b), $T = 0.12$ (c), $T = 0.1$ (d), $T = 0.01$ (e). The final panel (f) shows the result of running normal coverage starting from the configuration of (e).

To compare this algorithm with the normal limited-range (area) coverage algorithm in [2], we conducted 100 simulations of the normal coverage algorithm and compared the cost function (1) for $T = 0$. Eight agents with sensing radius $R = 2$ were uniformly and randomly distributed over Q. Over the 100 runs, the minimum area covered was 93.5% of the total area, the maximum was 96.4% and the mean was 95.9%. The limited-range DA algorithm had a final coverage area of 96.1%. For the particular ϕ in Figure 2, the normal (area) coverage simulations show how the final cost has dependence on initial conditions. However, the distributed DA algorithm converged to the same cost over many random initial conditions.

Next, we present a simulation of the limited-range DA algorithm using the mixed centroidal-area sensing function (5). The total number of agents is $N = 6$ and the square region Q has length 10 per side. We will demonstrate the performance of the DA algorithm versus a normal Lloyd-type gradient descent found in [2] as sensing radius decreases.

We first consider the limiting case where all agents can sense the entire region Q. We ran the

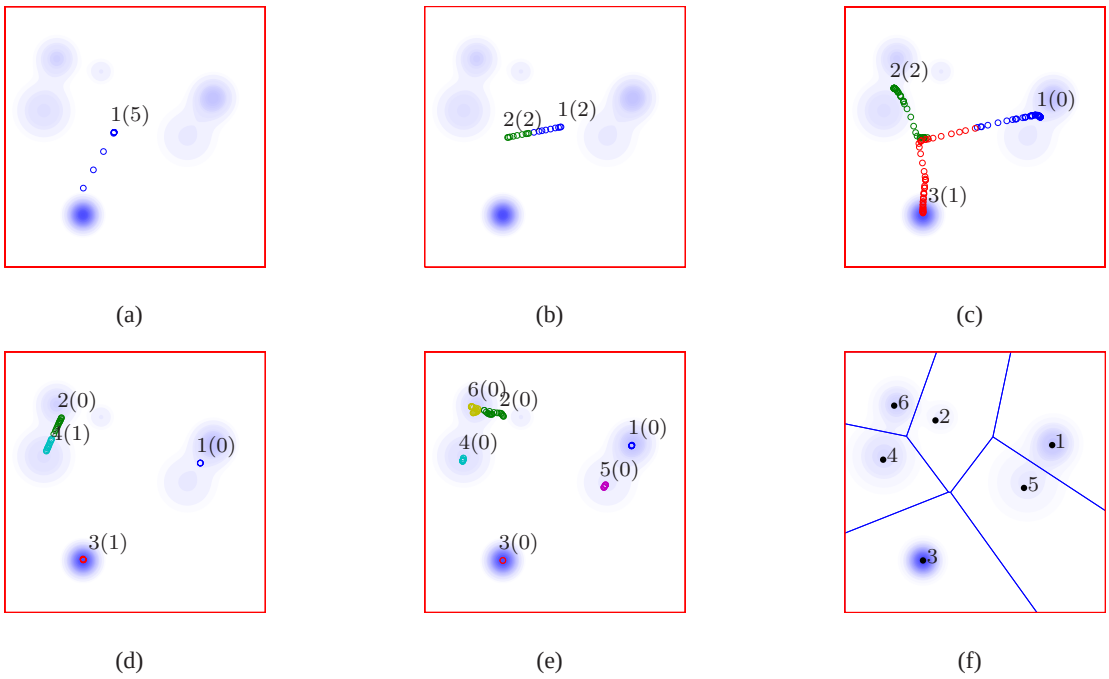


Fig. 3. A typical run of the limited-range DA algorithm. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). The temperature begins at $T = 20$, then decrease: $T = 13.1$ (a), $T = 12.5$ (b), $T = 2.4$ (c), $T = 2.1$ (d), $T = 0.5$ (e). The final panel (f) shows the result of running the normal gradient-descent coverage starting from the configuration of (e).

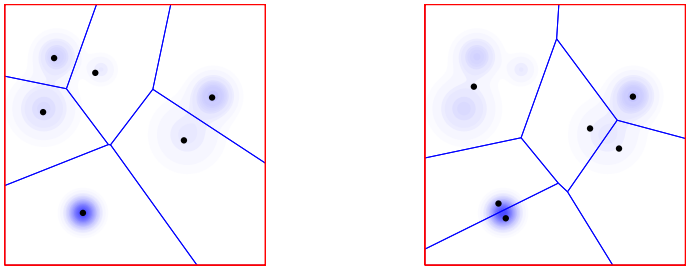


Fig. 4. A comparison between the best-case performance of the Lloyd-like gradient decent (left) with a cost of 10.64, to the worst-case (right) with a cost of 18.19.

DA algorithm once, since the initial condition does not influence the outcome of the convergence. For comparison, we ran 50 simulations starting from random initial conditions of the Lloyd-like algorithm. The DA algorithm converges to the optimal cost of 10.64 *regardless of initial condition* in Figure 3. On the other hand, the Lloyd-like gradient descent achieved this final configuration only 4 of the 50 tries. The worst simulation converged to a final cost of 18.19, see Figure 4.

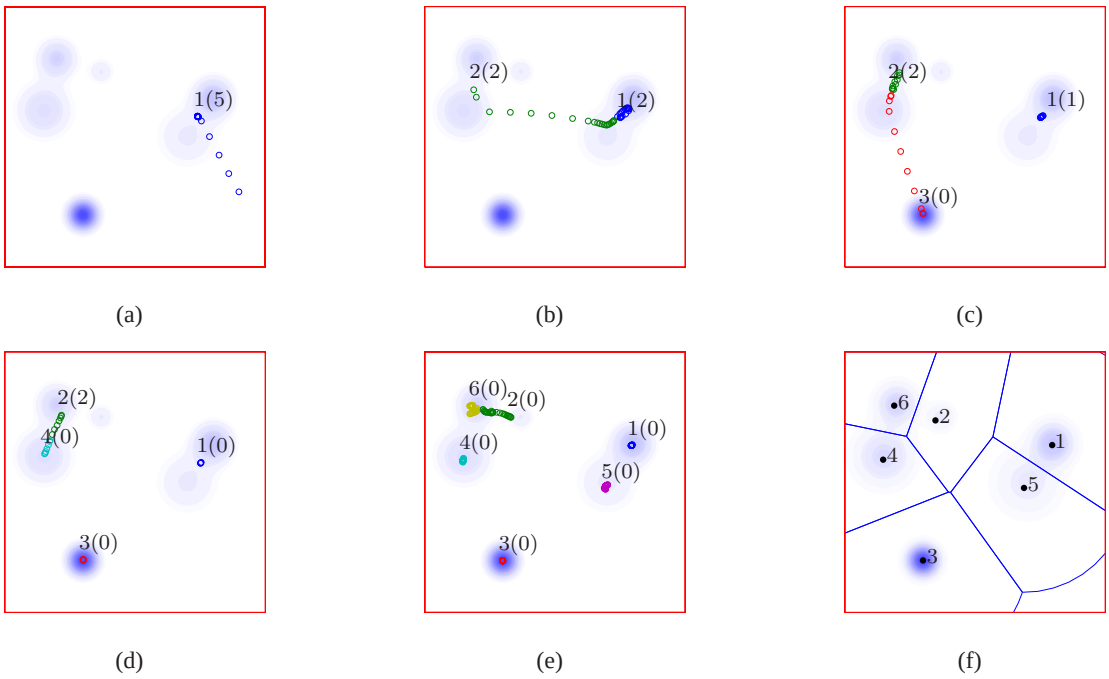


Fig. 5. A typical run of the limited-range DA algorithm with $R = 4$. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). The temperature begins at $T = 20$, then decrease: $T = 2.2$ (a), $T = 2.1$ (b), $T = 2.0$ (c), $T = 1.9$ (d), $T = 0.1$ (e). The final panel (f) shows the result of running the normal gradient-descent coverage starting from the configuration of (e).

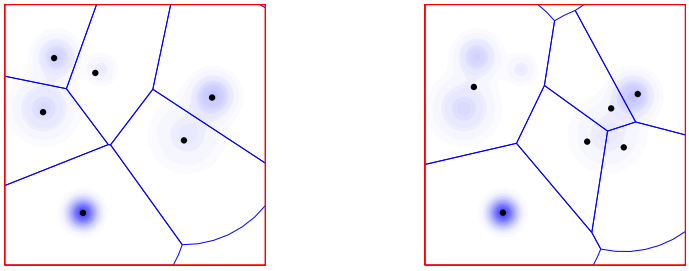


Fig. 6. A comparison between the best-case performance of the Lloyd-like gradient decent (left), to the worst-case (right). The sensing radius for all agents is $R = 4$.

The average final cost of the 50 simulations was 15.4.

Now we consider smaller sensing radii to see how the DA algorithm performs. A sensing range of $R = 4$ for each agent was sufficiently large for the DA algorithm to converge to the final configuration shown in Figure 5 regardless of initial position. Next, 50 trials of the Lloyd-like gradient descent were computed. The DA algorithm converges to the minimum cost of -277.5

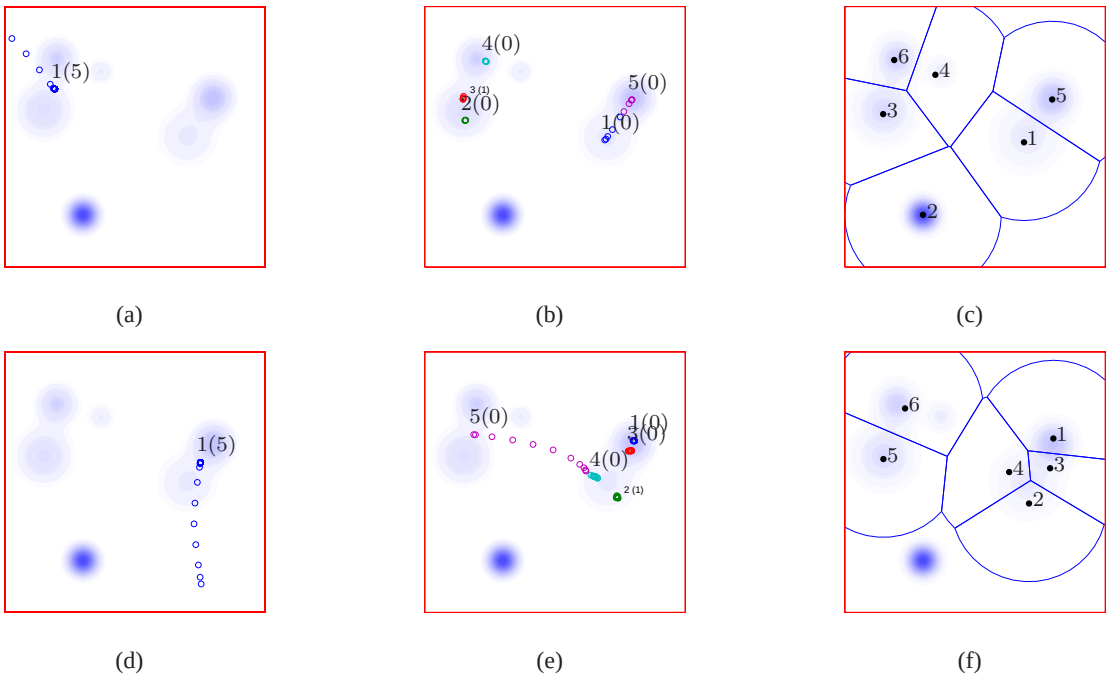


Fig. 7. Two runs of the limited-range DA algorithm with $R = 3$. In (a)–(c), the temperature begins at $T = 20$ and decreases: $T = 2.2$ (a), $T = 0.8$ (b), with a final configuration in (c). Similarly in (d)–(f), the temperature begins at $T = 20$ and decreases: $T = 2.0$ (d), $T = 0.5$ (e), with a final configuration in (f).

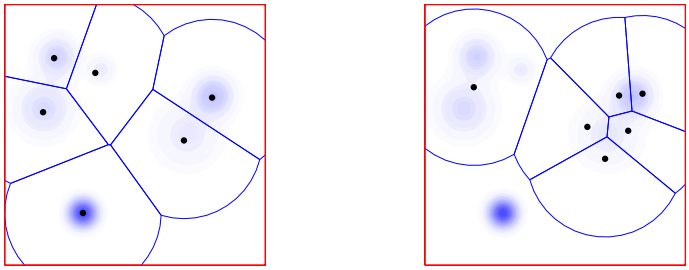


Fig. 8. A comparison between the best-case performance of the Lloyd-like gradient decent (left), to the worst-case (right). The sensing radius for all agents is $R = 3$.

regardless of initial position whereas only 3 of the 50 trials of the Lloyd-like gradient descent achieve that cost. The worst case converged to a cost of -270.3 with a mean of -272.3. The cost values are negative since the metric (5) has the $-R_i^2$ term. When inserted into the distortion function (1), this creates large negative quantities.

We further reduce the sensing radius to $R = 3$, and perform similar trials. Due to the smaller

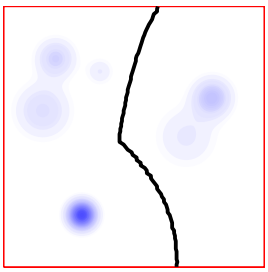
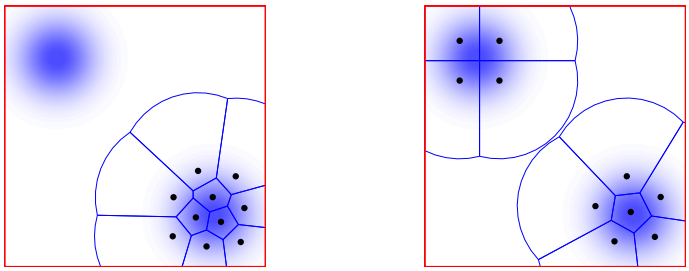


Fig. 9. For initial conditions of the limited-range DA algorithm that start to the left of the thick black line, simulations converge to the optimal solution.



(a) (b)

Fig. 10. A comparison between the best-case performance of the Lloyd-like gradient decent (left), to the worst-case (right) for the symmetric Gaussian scenario. The sensing radius for all agents is $R = 3$.

sensing radius, initial conditions begin to influence the outcome of the DA algorithm. For this particular choice of ϕ, we have the two possible outcomes shown in Figure 7. The better outcome of the DA attains a final cost of -151.5, while the worse outcome reaches a final cost of -110.5. Over the 50 random Lloyd-like gradient descent simulations, only 2 reach the configuration shown in 8, which is also 7 (c). The worst case of all 50 trials was a cost of -103.5, with an average of -135.3.

Further analysis of this scenario, however, demonstrates that the limited-range DA algorithm still has an advantage over a normal gradient descent algorithm. Figure 9 shows the set of initial conditions for which the limited-range DA algorithm converges to the best solution. Note that over half of the possible initial condition locations leads to the optimal solution while only 4% of the Lloyd-like gradient descent simulations achieved the same final cost.

The limited-range DA algorithm may have decreased performance versus a normal gradient-descent algorithm. If sensing range is not large enough, as was observed in the previous example,

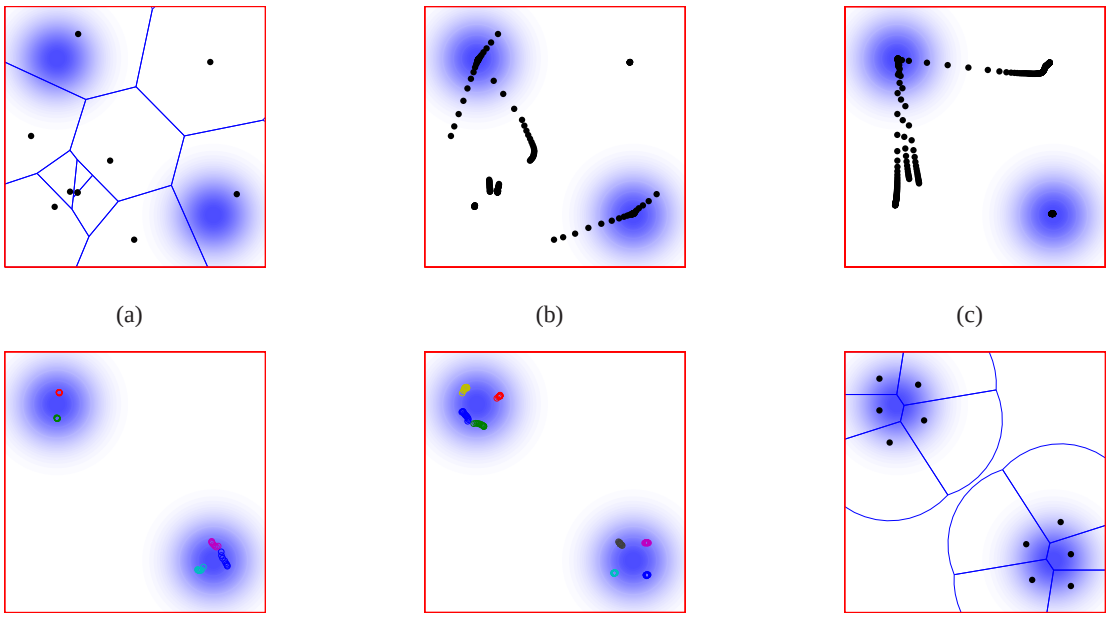


Fig. 11. A demonstration of a heating and cooling cycle with $R = 3$. Agents perform area-maximizing coverage (a), then agents run the limited-range DA algorithm for high temperature, $T = 20$, (b)–(c). Finally, agents perform the usual limited-range DA algorithm in (d)–(f).

the DA algorithm may fall into a local minimum. Consider the distribution shown in Figure 10, where there are two equal Gaussians symmetrically placed at opposite corners of Q. Almost every simulation of the limited-range DA algorithm results in a final configurations like 10(a), or its mirror image. This occurs because the DA algorithm begins with only one agent, and this agent moves towards the nearest Gaussian that it senses and stays there. Then, future phase changes result in only adding more agents around the same Gaussian.

On the other hand, over 50 trials of the Lloyd-like gradient descent with similar initial conditions as before, we see an improved statistic. Only 18 of the 50 simulations fell into the worst-case minima of Figure 10(a). However, none of the simulations were able to converge to the best configuration, which is having 5 agents located around each Gaussian.

A possible way to address this problem of the limited-range DA algorithm is to consider a heating and cooling cycle. Agents can deploy over Q using an area-maximizing technique. Thus, agents will tend to move away from each other and cover all of Q, as shown in Figure 11(a). Then, the limited-range DA algorithm is run with a high temperature. This forces agents to

collect together about denser parts of Q, shown in Figure 11(b)–(c). Finally, the usual limited-range DA coverage is run, causing agents to split evenly over the important areas of Q, as in Figure 11(d)–(f). Note, however, that the communication connectivity requirement must be modified so that an agent can communicate with any other agent in Q for this solution to work.

VII. Conclusions

We have introduced a limited-range and distributed implementation of the DA algorithm developed by Rose, and applied it to the coverage problem. We developed limited-range results that extend those in [8] and [9]. When the sensing radius is as large as the diameter of Q, this algorithm becomes the normal DA algorithm of Rose. While the limited-range DA algorithm is able to outperform a Lloyd-like gradient descent algorithm in many cases, the algorithm has its limitations as sensing range decreases. Consideration of a heating and cooling cycle produces improved results, but it is still an ad hoc solution to the underlying problem.

VIII. Acknowledgments

The authors would like to thank Jorge Cortes and Francesco Bullo for initial discussions on the use of DA in coverage control algorithms.

References

[1] R. M. Murray, Ed., *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics and Systems*. Philadelphia, PA: SIAM, 2003.

[2] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimisation & Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[3] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *International Conference on Distributed Autonomous Robotic Systems (DARS02)*, Fukuoka, Japan, Jun. 2002, pp. 299–308.

[4] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[5] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *IEEE Conf. on Decision and Control*, December 2005, pp. 2542–2547.

[6] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, August 2006.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[8] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 80, no. 11, pp. 2210–2239, 1998.

[9] P. Sharma, S. Salapaka, and C. Beck, "A scalable deterministic annealing algorithm for resource allocation problems," in *American Control Conference*, June 2006, pp. 3092–3097.

[10] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based coordination of autonomous swarms," *Automatica*, vol. 42, no. 7, pp. 1107–1119, July 2006.

[11] N. A. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann Publishers, 1997.