2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Coalition formation and motion coordination for optimal deployment

Michael Ouimet Jorge Cortés

***Abstract*— This paper presents a distributed algorithmic solution to achieve network configurations where agents cluster into coincident groups that are distributed optimally over the environment. The motivation for this problem comes from spatial estimation tasks executed with unreliable sensors. We propose a probabilistic strategy that combines a repeated game governing the formation of coalitions with a spatial motion component governing their location. We establish the convergence of the agents to coincident groups of a desired size in finite time and the asymptotic convergence of the overall network to the optimal deployment, both with probability 1. The algorithm is robust to agent addition and subtraction. From a game perspective, the algorithm is novel in that the players' information is limited to neighboring clusters. From a motion coordination perspective, the algorithm is novel because it brings together the basic tasks of rendezvous (individual agents into clusters) and deployment (clusters in the environment).**

I. Introduction

This paper is motivated by optimal spatial sampling problems under possibly failing communications. Consider a group of mobile robotic sensors that take point measurements of a random field over an environment and relay them back to a data fusion center. Assume that because of the features of the medium and the limited agent communication capabilities, it is known that only a fraction of these packets will arrive at the center, but it is not a priori known which ones will. Given that some sensors are not working and their identity is unknown, a reasonable strategy consists of grouping sensors together into clusters so that the likelihood of obtaining a measurement from the position of each cluster is higher. In this paper, our aim is to design a distributed algorithm that makes the network autonomously create groups of a desired size such that (i) members of each individual group become coincident, and (ii) the groups deploy in an optimal way with regards to the spatial estimation objective.

Literature review: There is an increasing body of research that deals with spatial estimation problems with possibly failing communications where packets are either received without corruption or not received at all, see e.g., [1], [2], [3], [4]. In particular, [4] shows that, for the problem motivating our algorithm design, the clustering strategy outlined above is not only reasonable but optimal in some cases: the configurations that maximize the expected information content of the measurements retrieved at the center correspond to agents grouping into clusters, and the resulting clusters being deployed optimally. Achieving such desirable configurations is challenging because of the spatially distributed nature of the problem and the agent mobility. Our technical approach combines elements of spatial facility location [5], rendezvous and deployment of multi-agent systems [6], and coalition formation games [7], [8]. From a game-theoretic perspective, our analysis of the coalition formation dynamics is novel because of the consideration of evolving and partial interaction topologies. From a motion coordination perspective, the novelty relies on the coupled dynamics between the coalition formation, the clustering, and the network deployment. Other works in cooperative control that employ game-theoretic ideas to solve tasks such as formation control, target assignment, self-organization for efficient communication, consensus, and sensor coverage include [9], [10], [11], [12].

Statement of contributions: The main contribution of the paper is the design and analysis of the Coalition formation and deployment algorithm. The aim of this synchronous, distributed strategy is to allow robotic agents to autonomously form groups of a given desired size while clustering together and deploying optimally in the environment. The deployment objective is encoded through a locational optimization function whose optimizers correspond to circumcenter Voronoi configurations. The algorithm design combines a repeated game component that governs the dynamics of coalition formation with a spatial motion law that determines how agents' positions evolve. In the game, agents seek to join the nearby coalition closest to the desired size. According to the motion coordination law, agents not yet in a well-formed coalition cluster together while agents in a coalition of the desired size also move towards the circumcenter of their Voronoi cell. Our main result, cf. Theorem IV.4, establishes that the executions of the Coalition formation and deployment algorithm converge in finite time to a configuration where agents are coincident with their own coalition and all coalitions are the desired size, and asymptotically converge to an optimal deployment configuration, each with probability 1. The algorithm does not require the agents to have a common reference frame, and is robust to agent addition and deletion. Finally, we illustrate these properties in simulations. Proofs are omitted for reasons of space and will appear elsewhere.

II. Preliminaries

We present some facts on computational geometry and coalition games that play a key role in the discussion.

A. Basic geometric notions

We denote by $\mathbb{R}$ and $\mathbb{Z}$ the sets of real and integer numbers, respectively. Let $\|\cdot\|$ be the Euclidean distance. Given a set $S \subset X$, let $\mathbb{F}(S)$ denote the collection of finite subsets of S and $S^c = X \setminus S$ its complement. Let $|S|$ denote the

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA, {miouimet,cortes}@ucsd.edu

cardinality of the set S. Let $\mathrm{vr} : \mathbb{R}^d \to \mathbb{R}^d$ be defined by $\mathrm{vr}(u) = u/\|u\|$ for $u \in \mathbb{R}^d \setminus \{0\}$, and $\mathrm{vr}(0) = 0$. We let $B(x,r) = \{p \in \mathbb{R}^d \mid \|x-p\| \leq r\}$. The circumcenter of a set of points P, denoted $\mathrm{CC}(P)$, is the center of the ball of minimum radius, denoted $\mathrm{CR}(P)$, which encloses all points in P. Next, we introduce the `get-together-toward-goal` function $\mathrm{gttg} : S \times \mathbb{F}(S) \times S \to S$ that will help us later to get a set of points P closer to each other while moving towards a goal q. Define

$$\mathrm{gttg}(p, P, q) = p + w_1 + w_2,$$

where we use the shorthand notation $P_0 = P \cup \{p\}$,

$$w_1 = \min\{\| \mathrm{CC}(P_0) - p\|, d_1(r)\} \ \mathrm{vr}(\mathrm{CC}(P_0) - p),$$
$$w_2 = \min\{\|q - (p + w_1)\|, d_2(r)\} \ \mathrm{vr}(q - (p + w_1)),$$

and $r = \mathrm{CR}(P_0)/\|q - \mathrm{CC}(P_0)\|$. Here, $d_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a continuous, increasing function on $(0, \infty)$ that satisfies

$$d_1(0) = 0, \quad \lim_{s\to\infty} d_1(s) = d_{\max}, \quad \lim_{s\to 0^+} d_1(s) = d_{\min},$$

for $d_{\max} > d_{\min} > 0$, and $d_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined by $d_2(s) = d_{\max} - d_1(s)$. Fig. 1 illustrates the gttg function.

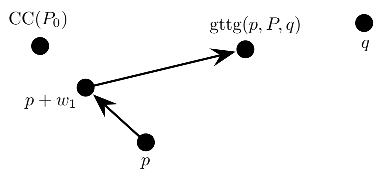


Fig. 1. Illustration of the action of the function gttg.

B. *Voronoi partitions and deployment objective*

Here, we introduce some computational geometric notions that play an important role in the formalization of the deployment problem. Given $Q \subset \mathbb{R}^d$ and a finite set of points $P = \{p_1, \dots, p_N\} \subset Q$, the Voronoi partition $V(P) = \{V_1(P), \dots, V_N(P)\}$ of Q is defined by

$$V_i(P) = \{q \in Q \mid \| q - p_i\| \leq \| q - p_j\|, \ \forall p_j \in P\}.$$

Note that $V_i(P)$, the Voronoi cell of p_i, is the set of points in Q closer to p_i than to any of the other points in P. The points p_i and p_j are (Voronoi) neighbors if the boundaries of their cells intersect. To compute the Voronoi cell of p_i, all that is required is the location of its neighbors in P. The work [13] introduces a procedure, that we term the ADJUST RADIUS strategy, which does the following: starting from $r = 0$, it repeatedly grows r until all Voronoi neighbors of p_i are guaranteed to be contained in $B(p_i, r)$.

Given a partition $\{W_1, \dots, W_N\}$ of Q, the disk-covering function $\mathcal{H}_{\mathrm{DC},N}$ is defined by

$$\mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N, W_1, \dots, W_N) = \max_{i\in\{i,\dots,N\}} \max_{q\in W_i} \|q - p_i\|_2.$$

The value of $\mathcal{H}_{\mathrm{DC},N}$ solves the following problem: cover the whole environment with balls centered at the points in $P = \{p_1, \dots, p_N\}$ with minimum common radius such that $W_i \subset B(p_i, r)$, for $i \in \{1, \dots, N\}$. For convenience, we use the notation $\mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N) = \mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N, V_1, \dots, V_N)$. Two properties are worth noting [6]: for a fixed configuration, the Voronoi partition is optimal among all partitions,

$$\mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N, V_1(P), \dots, V_N(P)) \leq \mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N, W_1, \dots, W_N),$$

and, for a fixed partition, the cells' circumcenters are optimal,

$$\mathcal{H}_{\mathrm{DC},N}(\mathrm{CC}(W_1), \dots, \mathrm{CC}(W_N), W_1, \dots, W_N) \leq \mathcal{H}_{\mathrm{DC},N}(p_1, \dots, p_N, W_1, \dots, W_N).$$

Under certain technical conditions, optimizing $\mathcal{H}_{\mathrm{DC},N}$ corresponds to minimizing the maximum error variance in the estimation of a random spatial field [14]. The deployment objective function that motivates our algorithm is given by

$$\mathcal{H}_{N,g}(p_1, \dots, p_N) = \frac{1}{\binom{N}{g}} \sum_{\{s_1,\dots,s_g\}\in C(N,g)} \mathcal{H}_{\mathrm{DC},g}(p_{s_1}, \dots, p_{s_g}), \quad (1)$$

where $C(N, g)$ denotes the set of unique g-sized combinations of elements in $\{1, \dots, N\}$. This function corresponds to the expected disk-covering performance of a network of N agents where only g of them are working and their identity is unknown. Optimizers of $\mathcal{H}_{N,g}$ correspond to grouping agents into coincident clusters of a specific size, say κ, that themselves are optimally deployed according to $\mathcal{H}_{\mathrm{DC},\lceil \frac{N}{\kappa} \rceil}$, see [4]. The cluster size κ is a function of N, g, and Q. For our problem, we assume that the optimal cluster size κ is known, and so forming coincident clusters of size κ and deploying these groups appropriately optimizes (1).

C. *Hedonic coalition games*

Hedonic coalition formation games [7] are N-player noncooperative games [15], [16] where players attempt to join/stay in preferable coalitions. Each player is hedonic because the utility it assigns to a given network coalition partitioning is only a function of its own coalition. Each player's action set is finite: it can stay in the current coalition or join another coalition. For a finite set of players $\mathcal{A} = \{1, \dots, N\}$, a finite coalition partition is a set $\Pi = \{S_k\}_{k=1}^K$, $K \in \mathbb{Z}_{\geq 1}$, that partitions $\mathcal{A}$. The subsets S_k are called coalitions. For player i and partition Π, let $S_\Pi(i)$ be the set $S_k \in \Pi$ such that $i \in S_k$. Agent i's preference is defined by an ordering $\succeq_i$ over the set $\mathfrak{S}_i = \{S \in \mathbb{F}(\mathcal{A}) \mid i \in S\}$. A coalition partition Π is called Nash stable if, for each $i \in \mathcal{A}$,

$$S_\Pi(i) \succeq_i S_k \cup \{i\}, \quad \forall S_k \in \Pi \cup \{\emptyset\}. \quad (2)$$

In coalition formation games, a player has full information about which coalitions all other players are in and may join any of them. This is in contrast to our scenario, where coalition information is only partial (and possibly incorrect), due to the limited capabilities of individual agents. Let us introduce definitions which help capture the spatially-limited nature of coalition information. We say that $(S_1, \dots, S_N)$ is a *consistent* coalition state if $i \in S_i$ and $S_j = S_i$, for each

$j \in S_i$, for each $i \in \mathcal{A}$. Note that for a consistent coalition state, $\{S_1, \dots, S_N\}$ reduces to a finite coalition partition of $\mathcal{A}$. Let $\tau_i \subseteq \mathcal{A}$ denote the set of agents whose coalition information i has access. Letting $S_0 = \emptyset$, the function best-set defines the players whose coalitions i most prefers to be a member of,

$$\begin{aligned}&\text{best-set}(\succeq_i, \{(k, S_k)\}_{k \in \tau_i}) = \\ &\{j \in \tau_i \cup \{0\} \mid S_j \cup \{i\} \succeq_i S_k \cup \{i\}, \forall\, k \in \tau_i \cup \{0\}\}.\end{aligned}$$

III. Problem statement

Consider a group of robotic sensors with unique identifiers $\mathcal{A} = \{1, \dots, N\}$ moving in a convex polygon $Q \subset \mathbb{R}^2$. Let p_i denote the location of agent i and $P = (p_1, \dots, p_N)$ denote the overall network configuration. We consider arbitrary agent dynamics, assuming each agent can move up to a distance $d_{\max} \in \mathbb{R}_{>0}$ within one timestep,

$$p_i(\ell + 1) \in B(p_i(\ell), d_{\max}), \quad \ell \in \mathbb{Z}.$$

Through either sensing or communication, we assume each agent i can get the relative position and identity of agents within distance $r_i \in \mathbb{R}_{>0}$. During the coalition formation process, agents can communicate with other agents within this radius. Agent i can adjust r_i but the cost of acquiring information is an increasing function of it. Inter-agent communication occurs instantaneously.

Given the problem scenario described in Section I, the network's objective is dual. On the one hand, agents want to cluster into groups of a predefined size κ. Equivalently, the network wants to self-assemble into $\lfloor \frac{N}{\kappa} \rfloor$ clusters of size κ, with possibly one additional cluster of size z, $0 \leq z < \kappa$, with $N = \lfloor \frac{N}{\kappa} \rfloor \kappa + z$. On the other hand, the resulting clusters should be positioned in the environment so as to minimize $\mathcal{H}_{\mathrm{DC}, \lceil \frac{N}{\kappa} \rceil}$. As discussed in Section II-B, such deployments correspond to optimizers of (1) for a class of spatial estimation problems with unreliable sensors. For convenience, we define a partition to be a *goal coalition partition* if the cardinality of m of its coalitions is κ, with the cardinality of the remaining one equal to z, if it exists.

A trivial solution to this problem would be to first elect $\lceil \frac{N}{\kappa} \rceil$ leaders and have each leader recruit $\kappa - 1$ followers. Then each group could rendezvous, and afterwards, the overall network would deploy. However, this method is neither distributed nor robust to agent failures. Our aim is to create a distributed algorithm that accomplishes the dual network objective in a robust and efficient way.

IV. Coalition formation and deployment algorithm

In this section, we solve the spatial deployment problem posed in Section III with the Coalition formation and deployment algorithm. This distributed, synchronous strategy specifies for each agent the dynamics of coalition formation and spatial motion. Section IV-A outlines the logic used by agents to determine which coalition to join as well as the supporting inter-agent communication and Section IV-B discusses how agents decide how to move depending on their coalition size and the deployment objective.

Before describing the dynamics, we begin with descriptions of the required memory of each agent i and appropriate initializations. The memory $\mathcal{M}_i$ of agent i is composed of

- the *coalition set* C_i. Elements of this set are of the form (j, p_j), i.e., identity and position of the member. For convenience, we set $(i, p_i) \in C_i$ and $C_0 = \emptyset$;
- the *communication radius* r_i at which the agent interacts with other agents not necessarily in its coalition set;
- the *neighboring set* $\mathcal{N}_i$ corresponding to agents within distance r_i, i.e., $(j, p_j) \in \mathcal{N}_i$ iff $p_j \in B(p_i, r_i)$;
- the *farthest-away radius* $\mathfrak{r}_i$, corresponding to the maximum distance to members of its coalition set.
- the *flag* `last`, which indicates if an agent belongs to the single final coalition not of size κ when $\lceil \frac{N}{\kappa} \rceil \neq \frac{N}{\kappa}$.

The operators $\mathrm{id}(\cdot)$ and $\mathrm{pos}(\cdot)$ extract identities and positions, respectively, from sets with elements of the form (i, p_i). Initialization requires a consistent coalition state $(\mathrm{id}(C_1), \dots, \mathrm{id}(C_N))$, $r_i \in \mathbb{R}_{\geq 0}$, and `last` = False.

A. Coalition formation game

The formation of coalitions evolves according to a simultaneous-action hedonic coalition game with partial information. Let us start with an informal description.

> *[Informal description]:* The agents' objective is to be in a κ-sized coalition. There are two rounds of communication per timestep. In the first one, each agent acquires information to determine if any neighboring coalition is more attractive than its current one. In the second one, the agents involved in a coalition change (either because they have decided to switch or because someone else decided to join their coalition) exchange information to update the coalition membership.

Next, we formally describe the hedonic coalition formation game. The agent i's preference ordering $\succeq_i$ over $\mathfrak{S}_i$ is

$$\begin{aligned}&\{S \in \mathfrak{S}_i \mid |S| = \kappa\} \succ \{S \in \mathfrak{S}_i \mid |S| = \kappa - 1\} \succ \dots \\ &\succ \{S \in \mathfrak{S}_i \mid |S| = 1\} \succ \{S \in \mathfrak{S}_i \mid |S| = \kappa + 1\} \succ \dots \\ &\qquad \succ \{S \in \mathfrak{S}_i \mid |S| = N\}. \quad (3)\end{aligned}$$

Next, we specify the two rounds of communication that take place per timestep. Agents who already are in a coalition of size κ do not actively take part in this process; they only respond to other agents' messages. First, agents execute the Best neighbor coalition detection strategy described as Algorithm 1. According to this strategy (cf. step 5), an agent that finds a neighboring coalition better than its own will decide to join it with probability given by

$$P\big(|C_i|, \kappa\big) = 1 - (1 - b)^{1/|C_i|} \quad \text{if } |C_i| \neq \kappa. \qquad (4)$$

If $|C_i| = \kappa$, the player i will surely not switch coalitions. The design parameter $b \in (0, 1)$ corresponds to the probability

Algorithm 1: BEST NEIGHBOR COALITION DETECTION

Executed by: Agents i with $|C_i| \neq \kappa$

Acquire $\mathcal{N}_i$ `% get location of neighbors`
Send $(\texttt{query}, r_i)$ at r_i to $\mathrm{id}(\mathcal{N}_i \setminus C_i)$
Receive $\mathrm{id}(C_j)$ from all $j \in \mathrm{id}(\mathcal{N}_i \setminus C_i)$
`% request/receive coalition sizes`
if $i \notin \text{best-set}(\succeq_i, \{k, \mathrm{id}(C_k)\}_{k\in\mathrm{id}(\mathcal{N}_i)})$ **then**
`% better coalitions exist`
  with probability $P\big(|C_i|, \kappa\big)$ **do**
    Set j^* from $\text{best-set}(\succeq_i, \{k, \mathrm{id}(C_k)\}_{k\in\mathrm{id}(\mathcal{N}_i)})$
    `% identify best coalition to join`
    if $j^* \neq 0$ **then** $r_i := \|p_{j^*} - p_i\|$
  end
end

that at least one agent in a non-κ coalition has the ability to act. The choice of b influences the rate of coalition changes.

Remark IV.1 (Justification for probabilistic actions) The probabilistic model for actions in (4) helps avoid deadlock situations that may result from the decentralized nature of the game. As an example, in a situation with two groups of size $\kappa-1$, all agents desire to join the other group. If this were the case, a group of size κ would never form. Instead, under (4), there is a positive probability $2b(1-b)$ that agents in only one of the groups act, breaking the deadlock. In contrast with a one-agent-acting-per-timestep policy, (4) allows multiple agents to switch coalitions simultaneously. •

Next, all agents execute the COALITION SWITCHING strategy described in Algorithm 2. This strategy builds on the input j^* provided to i by the BEST NEIGHBOR COALITION DETECTION strategy. Agents with $j^* \neq i$ switch coalitions. If $j^* = 0$, i forms its own coalition. Otherwise, i interacts with agent j^* to join its coalition. The strategy updates coalition memberships and the communication radii required to determine the position of other members so that the coalition state is consistent after the switches have occurred.

B. Motion control law

Here, we describe how agents move at each timestep, beginning with an informal description:

> *[Informal description]:* At each timestep, agents adjust their communication radius and move. Both of these actions are dependent on the size of their coalition. Agents not yet in a coalition of size κ increase their radius to improve the chances of finding a better coalition and move towards their coalition members. Agents in a coalition of size κ adjust their radius to ensure they can calculate their Voronoi cell and move towards both their coalition members and the circumcenter of their cell.

Formally, the RADIUS ADJUSTMENT AND MOTION strategy is described as Algorithm 3. Its interaction with the coalition formation dynamics is described in steps 10-16, which

Algorithm 2: COALITION SWITCHING

Executed by: All agents i

if $j^* \neq i$ **then**
  Send $(\texttt{leave}, i)$ at $\mathfrak{r}_i$ to $\mathrm{id}(C_i)$
  if $j^* \neq 0$ **then**
    Send $(\texttt{join}, i, r_i)$ at r_i to j^*
  end
  `% alert old and new coalitions`
end
$M := \{k \in \mathcal{A} \mid i \text{ received } \texttt{join} \text{ from } k\}$
`% agents relying on i to aid switching`
foreach $m \in M$ **do** Send $(\texttt{join}, m, r_m)$ to $\mathrm{id}(C_i)$
`% alert coalition members of m via r_i`
$L := \{k \in \mathcal{A} \mid i \text{ received } \texttt{leave} \text{ from } k\}$
$J := \{k \in \mathcal{A} \mid \text{an } m \in \mathrm{id}(C_i) \text{ got } \texttt{join} \text{ from } k\}$
`% agents leaving/joining i's coalition`
$\mathrm{id}(C_i) := (\mathrm{id}(C_i) \cup J) \setminus L$ and $r_i := \mathfrak{r}_i + \max\{r_j\}_{j\in J}$
`% update current coalition and radius`
foreach $m \in M$ **do** Send $(r_i, \mathrm{id}(C_i))$ at r_m to m
`% update agents joining i's coalition`
if $j^* \neq i$ **then**
  if $j^* = 0$ **then**
    $C_i = \{(i, p_i)\}$ `% form a new coalition`
  else
    $\mathrm{id}(C_i) := \mathrm{id}(C_{j^*})$ and $r_i := \|p_{j^*} - p_i\| + r_{j^*}$
    `% update coalition and radius`
  end
end
if $J \neq \emptyset \vee j^* \neq i$ **then**
  Acquire $\mathcal{N}_i$, $\mathrm{pos}(C_i)$
  $j^* := i$ `% reset switching variable`
end

governs the set of agents that a robot not yet in a κ-sized coalition may interact with. The next result ensures that the agent communication radius is kept at the smallest value guaranteeing a successful formation of coalitions.

Lemma IV.2 *For each $i \in \mathcal{A}$ such that $|C_i| \neq \kappa$, let k_i be the closest agent which is in a coalition different from i's with size different from κ. Let $r_i(P, (C_1, \dots, C_N)) = \|p_i - p_{k_i}\|$. For consistent coalition states not corresponding to a goal coalition partition, such radii guarantee that at least one agent has an incentive to switch coalitions. Moreover, if the radii of these agents were set according to any other function r'_i with $r'_i(P, (C_1, \dots, C_N)) < r_i(P, (C_1, \dots, C_N))$ for some i and P, then this property is no longer guaranteed.*

Remark IV.3 (Voronoi cell computation) In the computation in Algorithm 3, Step 5, the coalition's circumcenter replaces the locations of all individual agents ensuring that all coalition members compute the same cell. However, this implies that the collection of cells computed by the coalitions is not a partition of the environment. This issue is resolved when all agents are coincident with their coalition. •

Algorithm 3: RADIUS ADJUSTMENT AND MOTION

Executed by: All agents i

if $|C_i| = \kappa \vee \texttt{last} = \text{True}$ **then**
  Update r_i with ADJUST RADIUS strategy
  Acquire $\mathcal{N}_i$
  $A_i := (\{\text{CC}(\text{pos}(C_i))\} \cup \text{pos}(\mathcal{N}_i)) \setminus \text{pos}(C_i)$
  $V_i := V_1(A_i)$ `% compute Voronoi cell`
  goal $= \text{CC}(V_i)$
else
  goal $= \text{CC}(\text{pos}(C_i))$
  $\mathfrak{C}_i := \{j \in \text{id}(\mathcal{N}_i) \mid |\text{id}(C_i)| = \kappa\}$
  if $\text{id}(\mathcal{N}_i \backslash C_i) \backslash \mathfrak{C}_i \neq \emptyset$ **then**
    $r_i := \min_{k \in \text{id}(\mathcal{N}_i \backslash C_i) \backslash \mathfrak{C}_i} \|p_k - p_i\| + 2d_{\max}$
    `% guarantees a neighbor after motion`
  else
    if $\text{id}(\mathcal{N}_i) = \mathcal{A}$ **then**
      $\texttt{last} := \text{True}$ `% one non-`κ` coalition`
    else
      $r_i := r_i + \delta$ `% increase radius`
    end
  end
end
foreach $j \in \text{id}(C_i)$ **do** $p_j := \text{gttg}(p_j, \text{pos}(C_i), \text{goal})$
`% compute next position`
$\text{pos}(C_i) := \{p_j\}_{j \in \text{id}(C_i)}$ `% update positions`
$\mathfrak{r}_i := \max_{p_j \in \text{pos}(C_i)} \|p_j - p_i\|$ `% recompute radius`

The COALITION FORMATION AND DEPLOYMENT ALGORITHM is the composition of Algorithms 1-3. This strategy does not require agents to share a common reference frame and is robust to agents joining or leaving the network provided that: (i) new agents alert the network by sending a `query` message, (ii) when an agent fails, the other members of its coalition detect this fact, (iii) when agents receive a `query` message they set $\texttt{last} := \text{False}$.

C. Convergence analysis

The next result states the convergence properties of the COALITION FORMATION AND DEPLOYMENT ALGORITHM.

Theorem IV.4 *Consider a network of N agents executing the* COALITION FORMATION AND DEPLOYMENT ALGORITHM. *The following holds,*

(i) *there exists a finite time after which the agents are in a goal coalition partition and each is coincident with its coalition members, with probability* 1*;*
(ii) *the network asymptotically converges toward the set of minimizers of $\mathcal{H}_{\text{DC},\lceil \frac{N}{\kappa} \rceil}$, with probability* 1.

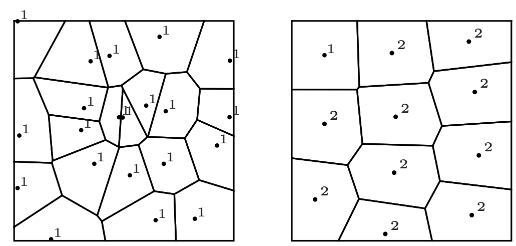
This result implies that agents may be stuck for some time in a different partition but, in finite time, they will reach the desired partition with probability 1. This can be traced back to the fact that, in the simplified coalition formation game where agents have both full information and action sets, the goal coalition partition is the only Nash stable partition.

V. SIMULATIONS

This section presents simulations of the COALITION FORMATION AND DEPLOYMENT ALGORITHM. We illustrate the convergence to a desired goal coalition partition and the achievement of the deployment task, the robustness against agent addition and subtraction, and the average coalition formation time as functions of N, k, and b. Regarding (4), in all simulations where b is constant, we have chosen $b = 0.5$. In all simulations, $\delta = d_{\max} = \frac{.2}{\sqrt{N}}$. We use the function

$$\phi(C_1, \ldots, C_N) = \frac{1}{N(\kappa - 1)} \sum_{i \in \mathcal{A}} ||C_i| - \kappa|, \qquad (5)$$

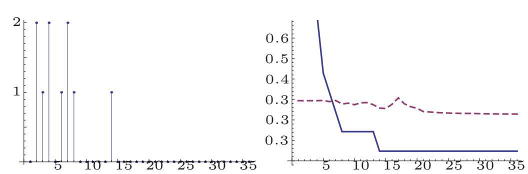
to illustrate the dynamics of coalition formation. This function is zero if and only if all agents are in κ-sized coalitions. Fig. 2 shows an execution of the COALITION FORMATION AND DEPLOYMENT ALGORITHM with 21 agents forming coalitions of size 2.



(a) Initial configuration (b) Final configuration

Fig. 2. Execution of the COALITION FORMATION AND DEPLOYMENT ALGORITHM with 21 agents and $\kappa = 2$. The network converges to a configuration where all agents are in correctly-sized coalitions and these coalitions are optimally deployed.

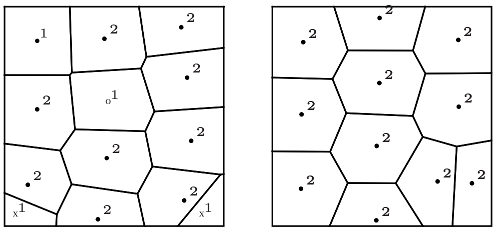
The network converges to both correctly sized groups and coalitions optimally deployed at their Voronoi cell's circumcenters. From Theorem IV.4, the final configuration optimizes $\mathcal{H}_{\text{DC},11}$. Fig. 3(a) shows the number of coalition switches at each timestep for the same run.



(a) (b)

Fig. 3. For the execution in Fig. 2, (a) shows the number of agents switching coalitions at each timestep, and (b) shows the evolution of ϕ (solid line) as defined in (5) and $\mathcal{H}_{21,20}$ (dashed line) as defined in (1).

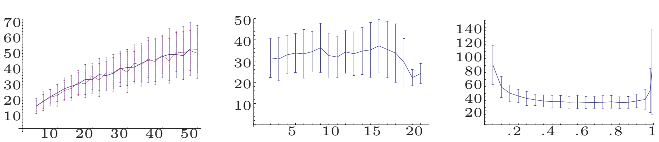
Many switches happen early, but decrease in frequency as agents form κ-sized coalitions. The evolution of ϕ depicted in Fig. 3(b) confirms this. This plot also shows the evolution of $\mathcal{H}_{N,N-1}$ that, in the language of Section II-B, corresponds to the situation where $N - 1$ of the sensors are working. This choice of function is motivated by the fact that, in one dimension, it is known that in such a case, forming coalitions of size 2 is optimal [4]. The bumps in the evolution of $\mathcal{H}_{21,20}$ in the plot occur when an agent with no nearby coalitions to join must increase its radius to it join a group far away from it. $\mathcal{H}_{21,20}$ temporarily increases while these agents get together. Fig. 4 illustrates the robustness of the COALITION FORMATION AND DEPLOYMENT ALGORITHM as well as convergence when $\lceil \frac{N}{\kappa} \rceil \neq \frac{N}{\kappa}$. After agents have achieved the final optimal configuration seen in Fig. 2(b), we let one agent fail and two new agents come into the picture. The agents adapt to the new network composition and optimally deploy according to the available resources.



(a) Initial configuration (b) Final configuration

Fig. 4. Execution of the COALITION FORMATION AND DEPLOYMENT ALGORITHM from the configuration in Fig. 2(b) where an agent has failed in the coalition marked as 'o' and two agents, marked as 'x', have been added. After these agent additions and subtractions, coalitions adapt and the network re-converges to an optimal deployment configuration.

Finally, Fig. 5 illustrates the dependency of the average number of timesteps required for all coalitions to form on N, κ, and b. Each point is the average of 200 runs, where the agents were initially randomly placed with uniform distribution in a unit square. The error bars correspond to plus and minus one standard deviation. Fig. 5(a) shows the average coalition formation convergence time for different N for cases of fixed $\kappa = 4$ and changing $\kappa = \lfloor \frac{N}{2} \rfloor$. In both cases, the completion time appears linear in N and each take a similar amount of time. The latter is confirmed in Fig. 5(b), which shows the average coalition formation convergence time for fixed $N = 20$ and varying κ. The coalition formation time is roughly equal for all desired coalition sizes, until nearly all agents are joining one coalition, which takes less time on average. Fig. 5(c) shows the average coalition formation



(a) (b) (c)

Fig. 5. (a) shows the average coalition formation time as a function of the number of agents N, for $\kappa = 4$ (dashed line) and $\kappa = \lfloor \frac{N}{2} \rfloor$ (solid line). (b) shows the average coalition formation time as a function of desired coalition size κ for $N = 20$ agents. (c) shows the average coalition formation time for 20 agents forming coalitions of size 4 as a function of b. In all plots, the error bars correspond to plus and minus one standard deviation.

time for 20 agents forming coalitions of size 4 for various b values. The completion time is roughly constant for values of b away from 0 and 1.

VI. CONCLUSIONS

Motivated by a spatial estimation problem, we have designed a synchronous, distributed algorithm for a network of robotic agents to autonomously deploy in groups over a region. Our strategy allows agents to autonomously form coalitions of a desired size, cluster together within finite time, and asymptotically reach an optimal deployment, with probability 1. The algorithm design combines a hedonic coalition formation game where agents only have partial information about other coalition memberships with motion coordination strategies for group clustering and deployment. The proposed solution is provably correct, does not rely on a common reference frame and is robust to agents joining or leaving the environment. Simulations illustrated these features along with the dependency of the average coalition formation time on N, κ, and b. Future work will be devoted to analytically characterizing the time complexity, as well as investigating δ policies which optimize the coalition formation process. We also plan to further explore the impact of noncooperative game-theoretic ideas in other motion coordination problems.

ACKNOWLEDGMENTS

This research was supported by NSF award CMMI-0908508.

REFERENCES

[1] S. C. Smith and P. Seiler, "Estimation with lossy measurements: jump estimators for jump systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 1453–1464, 2003.

[2] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[3] V. Gupta, A. F. Dana, J. Hespanha, R. M. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807–1819, 2009.

[4] J. Cortés, "Deployment of an unreliable robotic sensor network for spatial estimation," *Systems & Control Letters*, 2010, submitted.

[5] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., ser. Wiley Series in Probability and Statistics. Wiley, 2000.

[6] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[7] A. Bogomolnaia and M. Jackson, "The stability of hedonic coalition structures," *Games and Economic Behavior*, vol. 38, no. 2, pp. 201–230, Feb 2002.

[8] S. Banerjee, H. Konishi, and T. Sonmez, "Core in a simple coalition formation game," *Social Choice and Welfare*, vol. 18, pp. 135–153, 2001.

[9] D. Gu, "A differential game approach to formation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 85–93, January 2008.

[10] J. Marden, G. Arslan, and J. Shamma, "Cooperative control and potential games," *IEEE Transactions on Systems, Man & Cybernetics. Part B: Cybernetics*, vol. 39, pp. 1393–1407, 2009.

[11] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game theoretic formulation," *ASME Journal on Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 584–596, 2007.

[12] W. Saad, Z. Han, A. Hjorunges, D. Niyato, and E. Hosain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, January 2011.

[13] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[14] R. Graham and J. Cortés, "Asymptotic optimality of multicenter Voronoi configurations for random field estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 153–158, 2009.

[15] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA: MIT Press, 1991.

[16] T. Basar and G. Oldser, *Dynamic Noncooperative Game Theory*. Academic Press, 1982.