2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Team-based Coverage Control of Moving Sensor Networks

Farshid Abbasi†, Afshin Mesbahi† and Javad Mohammadpour†

***Abstract*— In this paper, the coverage optimization problem for mobile sensing networks is examined from a team-based perspective. The objective is to arrange robots in a given environment to minimize the serving cost based on a given density function, which defines the probability of events in the environment. A *team* concept is introduced here to allow adjusting to the changes in the environment or the assigned task(s) in a collaborative way as a team. Firstly, an optimization problem is solved in the team level in order to assign a sub-region to each team. The region assigned to each team is then divided among its members by minimizing a locational cost. A distributed communication algorithm is also given, in which each team member, e.g., robot can access the information of its neighbors in order to compute the associated region. Finally, numerical simulation results are given to demonstrate the effectiveness of the presented approach. The results indicate that agents in both teams- and agents-level reach the optimal configuration considering the given density function.**

I. Introduction

Due to the advantages of the distributed systems such as reliability, speed and economics over the centralized systems, the distributed deployment algorithms have been proposed for the workload sharing and partitioning tasks [1], [2], [3]. To this aim, each robot needs to only exchange information collected by its sensors, e.g., position and velocity, with other agents and negotiate its scheduled task with a number of other agents. Then, each robot locally generates an appropriate control action using information gathered from its neighbors [4], [5]. For mobile sensing networks, a distributed control strategy has been proposed in [6] to equally divide an assigned area into subregions, where each robot is able to obtain locations of its neighbors through, e.g., adjust-communication-radius algorithm. Then, it computes the associated Voronoi cell and moves toward the centroid of its Voronoi cell obtained based on the gradient descent method that gives the optimal solution for equal partitioning problem [7], [8].

The objective function defined for the partitioning problem represents a performance measure of the agents. In essence, the solution to this optimization problem maximizes the performance index in a way that the agents are deployed in the most optimum way. In other words, the distance from each agent to the points inside its allocated area is minimized in the coverage problem that results in a higher sensing performance for each agent [6]. This algorithm has been extended for power diagrams so that not only equitable partitions can be obtained in a spatially distributed manner, but also equitable and median Voronoi diagrams are achieved [9].

The existing approaches for the coverage control are based on the assumption that all robots belong to a single team. However, this assumption is not realistic in many real-world applications, as the agents may differ from dynamics and communication perspective. A multi-robot system can generally be considered as a homogeneous or heterogeneous system that translates to deploying a variety of robots with respect to the assigned task. In the present work, an alternative coverage strategy is proposed that aims at taking the differences in the robots dynamics into account by offering the *team concept*. Each robot might team up with other ones based on its assigned task, associated dynamics or embedded communication system. Therefore, it would be possible to improve reliability and flexibility of the deployment algorithm.

The present work introduces a new team-based coverage control scheme which can handle different scenarios in heterogeneous systems of robots. The presented approach addresses the problem of the agents deployment by considering teams of robots instead of evaluating each agent individually. The main problem can be defined as a two-level optimization problem, where one problem is defined inside each team and another is defined among the teams in the overall coverage space. The agents will move towards the local minima until the optimum configuration is achieved. In the proposed approach, firstly, a local minimum to the deployment problem is obtained in the team level. Then, a second optimization problem is solved to guarantee the convergence of the agents to their optimum location generating their respective Voronoi cells inside the teams. By considering the nucleus as the associated Voronoi centroid of each team, the optimization problem at the team level is defined to maximize the performance of the team.

The agents in each team can be classified into two groups of interior members and members on the boundaries based on whether or not they share a boundary with the agents belonging to other teams. Each interior member is able to compute its own Voronoi cell by only knowing the location of its neighbors. The location of the neighbors is obtained through the modified version of the adjust-communication-radius algorithm which has been presented in [6]. However, this does not hold true for the members on the boundaries. These members need not only the location of neighbor robots in their associated team, but also the nucleuses of the neighboring teams with their team Voronoi cells. The agents on the boundaries can obtain the nuclei of neighboring teams through the use of the proposed communication algorithm

† College of Engineering, The University of Georgia, Athens, GA 30602, E-mail: javadm@uga.edu

978-1-4673-8682-1/$31.00 ©2016 AACC

(see Section IV).

The remainder of this paper is structured as follows. Definitions and the problem statement are provided in Section II. Section III introduces a two-level optimization problem. An extended version of the classic Lloyd algorithm is applied to ensure the coverage properties. The agents communication and protocols are discussed in Section IV. Section V presents numerical simulation results to illustrate the team-based partitioning.

A. Notations

We use $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_+$ to denote the sets of positive natural, real, and nonnegative real numbers. The closed circle centered at $c \in \mathbb{R}^2$ with radius $r \in \mathbb{R}_+$ is defined by $B(c, r) := \{x \in \mathbb{R}^2 | \|x - c\| \leq r\}$. We define Q as a convex polytope in $\mathbb{R}^2$ and let $\mathscr{Q} = \{Q_1, Q_2, \dots, Q_t\}$ be a *partition* of Q as a collection of t closed subsets with disjoint interiors. Moreover, the so-called *distribution density function* is denoted by φ where $\varphi : Q \to \mathbb{R}_+$ represents the probability of some phenomenon occurring over space Q. The function φ is assumed to be measurable and absolutely continuous. The Euclidean distance function is denoted by $\| \cdot \|$ and $|Q|$ represents the Lebesgue measure of convex subset Q. The vector set $\mathcal{P}_t = (p_{t1}, p_{t2}, \dots, p_{tn_t})$ is the location of n_t agents belonging to t^{th} team moving in the space Q_t. As expected, the sensing performance of the agents decay as we move away from their location, and hence, sensing performance can be evaluated as a function of distance from the agent, i.e., $f(\|q - p_{tm}\|)$, where $q \in Q$.

II. A Team-based Optimization Scheme

A. Optimization problem

In the literature, the locational optimization function is presented in the following form that is translated to maximizing the sensing performance

$$\mathcal{H}(P, Q) = \sum_{i=1}^{N} \int_{\mathcal{W}_i} f(\|q - p_i\|)\varphi(q)\mathrm{d}q, \tag{1}$$

where for n teams, $N = \sum_{t=1}^{n} n_t$, and P is the set of all agents. It is assumed that i^{th} agent is assigned to the region $\mathcal{W}_i$ and the cost function $\mathcal{H}$ is minimized by finding the optimum locations of the agents and their assigned regions $\mathcal{W}_i$ whose union is Q. In this context, all the agents are assigned over the space no matter how the agents can collaborate or coordinate with each other in a more local platform. This is addressed in the present work by introducing a team-based partition of the agents that considers agents as a collection of multiple teams pursuing their assigned task or objective.

B. Voronoi partitions

The main objective of this work is to adopt the team concept in the agents deployment and partitioning framework. To do so, we first need to define an optimization problem that can handle not only the deployment and partitioning tasks inside teams but also the partitioning inside the defined polytope Q. This can be done by breaking the optimization problem into two interconnected functions in a way that the solution to each problem represents the optimum configuration of the teams and their associated agents.

To start with, we define the set of teams by $\mathcal{L} = (l_1, l_2, \dots, l_n)$ where each $l_t, t = 1 : n$, represents the nucleus of team t that is a function of the agents position in the associated team $l_t = g(p_{t1}, p_{t2}, \dots, p_{tn_t})$. The function dependency of l_t on the position of the agents is discussed later. Now, we can partition the polytope Q into a set of Voronoi cells $\mathcal{V}(\mathcal{L}) = \{V_1, V_2, \dots, V_n\}$ considered as the optimal partitioning for a set of agents with fixed locations at a given space as

$$V_t = \{q \in Q | \; \|q - l_t\| \leq \|q - l_s\|\}. \tag{2}$$

The obtained Voronoi cells associated with the nuclei of the teams are then considered as the convex polytopes set to deploy their associated agents. Therefore, the sub-partitions are defined on the basis of the Voronoi cells V_t obtained from the team level partitioning. The Voronoi partitions $\mathcal{V}_t(\mathcal{P}_t) = \{V_{t1}, V_{t2}, \dots, V_{tn_t}\}$ generated by the agents $(p_{t1}, p_{t2}, \dots, p_{tn_t})$ belonging to t^{th} team are defined as

$$V_{tm} = \{q \in V_t | \; \|q - p_{tm}\| \leq \|q - p_{tr}\|\}, \tag{3}$$

where p_{tm} denotes the location of m^{th} agent in t^{th} team such that $m \in \{1, \dots, n_t\}$.

We recall the basic characteristics of the Voronoi partitions including their associated mass, centroid and polar moment of inertia as [6]

$$\begin{gathered} M_{V_{tm}} = \int_{V_{tm}} \varphi(q)\mathrm{d}q, \; C_{V_{tm}} = \frac{1}{M_{V_{tm}}} \int_{V_{tm}} q \, \varphi(q)\mathrm{d}q, \\ J_{V_{tm}, p_{tm}} = \int_{V_{tm}} \|q - p_{tm}\|^2 \varphi(q)\mathrm{d}q. \end{gathered} \tag{4}$$

In addition to the defined parameters for the Voronoi cells of each agent inside the teams, we also need to define the characteristics of the teams's Voronoi cells. According to the definition of the Voronoi partitions, it can be deduced that

$$M_{V_t} = \sum_{m=1}^{n_t} M_{V_{tm}}, \; C_{V_t} = \frac{1}{M_{V_t}} \int_{V_t} q \, \varphi(q)\mathrm{d}q. \tag{5}$$

The nucleus of the team that is a function of the agents position is defined as

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} p_{tm}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{6}$$

As described earlier, the nucleus is a representative of the agents position in the team and can be considered as the collective position of the agents for drawing the Voronoi diagram of the teams V_t. The agents formation is called centroidal Voronoi configuration if it satisfies $l_t = C_{V_t}$, $p_{tm} = C_{V_{tm}}$ for both interior agents and nuclei of the teams.

C. Local optimization problem

The deployment task in the presented team-based scheme can be addressed by solving a two-level optimization problem. At first, we need to define an optimization problem, whose solution can guarantee the optimum configuration and partitioning of the convex polytope Q while applying the nuclei of the teams as functions of the agent position. Hence, the following cost function is considered to determine the nuclei of the teams

$$\mathcal{G}(\mathcal{L}, \mathcal{Q}) = \sum_{t=1}^{n} \int_{Q_t} \|q - l_t\|^2 \varphi(q) \mathrm{d}q, \quad (7)$$

where the sensing performance is considered as $f(\|q-l_t\|) = \|q - l_t\|^2$. The solution to (7) gives the local minimum to the deployment problem in the teams level. Once the optimum partitioning is achieved at the teams level, we need to define a second optimization problem, whose solution can guarantee the convergence of the agents inside the teams to their optimum location and by generating their respective Voronoi cells. We define a set of polygons as $\mathcal{Q}_t = \{Q_{t1}, Q_{t2}, \ldots, Q_{tn_t}\}$ with disjoint interiors, whose union is V_t. The following cost function is defined for each team as a function of the position of its associated agents

$$\mathcal{G}_t(\mathcal{P}_t, \mathcal{Q}_t) = \sum_{m=1}^{n_t} \int_{Q_{tm}} \|q - p_{tm}\|^2 \varphi(q) \mathrm{d}q. \quad (8)$$

It is proven that among different partitioning schemes the Voronoi partitions are optimum in the sense of minimizing both individually defined cost functions (7) and (8) [6]. Hence, for a given set of nuclei $\mathcal{L} \in Q$, agents position $\mathcal{P}_t \in V_t$, a *partition* $\mathcal{Q}$ of Q and a *partition* $\mathcal{Q}_t$ of V_t, it satisfies

$$\mathcal{G}(\mathcal{L}, \mathcal{V}(\mathcal{L})) \leq \mathcal{G}(\mathcal{L}, \mathcal{Q}), \quad (9)$$

$$\mathcal{G}_t(\mathcal{P}_t, \mathcal{V}_t(\mathcal{P}_t)) \leq \mathcal{G}_t(\mathcal{P}_t, \mathcal{Q}_t). \quad (10)$$

This concludes that the Voronoi cells represent the optimum partitioning. Furthermore, for any $\mathcal{L}' = (l'_1, l'_2, \ldots, l'_n) \in Q$ and $\mathcal{P}'_t = (p'_{t1}, p'_{t2}, \ldots, p'_{tn_t}) \in V_t$ satisfying $\|l'_t - C_{V_t}\| \leq \|l_t - C_{V_t}\|$ and $\|p'_{tm} - C_{V_{tm}}\| \leq \|p_{tm} - C_{V_{tm}}\|$, respectively, we have

$$\mathcal{G}(\mathcal{L}', \mathcal{Q}) \leq \mathcal{G}(\mathcal{L}, \mathcal{Q}), \quad (11)$$

$$\mathcal{G}_t(\mathcal{P}'_t, \mathcal{Q}_t) \leq \mathcal{G}_t(\mathcal{P}_t, \mathcal{Q}_t). \quad (12)$$

In other words, the given cost function is minimized when agents are at the centroids of their corresponding Voronoi cells.

III. Two-Step Optimization Problem

The problem of deploying teams of robots is broken into two optimization problems. The first function to be minimized represents the cost function associated with partitioning the main space into partitions related to the teams of agents. This is followed by another optimization problem that is solved inside each team. The solution to the second optimization problem results in deploying the agents in an optimum way inside the teams. To achieve this, we assume that both functions $\mathcal{G}$ and $\mathcal{G}_t$ in (7) and (8) are smooth and continuous over their regions. Then, the derivatives of the cost functions are obtained as [7]

$$\frac{\partial \mathcal{G}}{\partial l_t} = M_{V_t}(l_t - C_{V_t}), \quad (13)$$

$$\frac{\partial \mathcal{G}_t}{\partial p_{tm}} = M_{V_{tm}}(p_{tm} - C_{V_{tm}}). \quad (14)$$

It can be seen that if the teams nucleus and agents position move to the centroid of their Voronoi cells, the local minimum is achieved. Accordingly, the critical partitions and points for $\mathcal{G}$ and $\mathcal{G}_t$ are called centroidal Voronoi partitions.

A. Equivalence of agents and teams level optimization

As discussed earlier, we need to first assign the optimum partitions to the teams of agents and then deploy the robots in the assigned regions to their associated teams. In this section, we show that these two optimization problems are interrelated. In fact, it is shown that the solution to the second optimization problem can lead to the solution to the first problem considering the given dependence of the nucleus on the agents position.

Theorem 3.1: If the agents are in the critical points of the function $\mathcal{G}_t$, then with respect to the defined teams nucleus l_t in (6), it is concluded that the function $\mathcal{G}$ is minimized.

Proof: According to (14), the function $\mathcal{G}_t$ is minimized when $p_{tm} = C_{V_{tm}}$. Assuming that all the agents inside the teams are in the centroid of their respective Voronoi cells, the nucleus of the teams can be found as

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} C_{V_{tm}}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \quad (15)$$

Substituting the centroid of the agents Voronoi in (4), we have

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} \int_{V_{tm}} q\ \phi(q) \mathrm{d}q / M_{V_{tm}}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \quad (16)$$

Since the density distribution function is positive on Q, each Voronoi cell has a nonzero measure, i.e., $M_{V_{tm}} \neq 0$ for $m = \{1, 2, \ldots, n_t\}$. Hence, we obtain

$$l_t = \frac{\sum_{m=1}^{n_t} \int_{V_{tm}} q\ \phi(q) \mathrm{d}q}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \quad (17)$$

Considering that the Voronoi cells share no interior space with each other and also due to the continuity of the density function ϕ over Q, it can be concluded that

$$l_t = \frac{\sum_{m=1}^{n_t} \int_{V_{tm}} q\ \phi(q) \mathrm{d}q}{M_{V_t}} = \frac{\int_{V_t} q\ \phi(q) \mathrm{d}q}{M_{V_t}} = C_{V_t}. \quad (18)$$

Therefore, it was shown that when agents are in their respective centroids, the nucleus of the team is also in the centroid of the team's Voronoi cell. ■

B. Team-based Lloyd algorithm

To minimize the cost functions $\mathcal{G}$ and $\mathcal{G}_t$ over time, an extension of Lloyd algorithm [6], [7] is proposed here. The proposed algorithm should consider updating the team Voronoi cells while evolving the positions and partitions of the agents inside the teams. According to Theorem 3.1, if the agents move towards the centroid of their Voronoi cells, the nucleus of each team also evolves towards its associated centroid. Hence, the two level optimization problem can be seen as the problem of assigning agents to their associated regions inside a Voronoi cell with changing boundaries. The following dynamics is enforced on the agents

$$\dot{p}_{tm} = u_{tm}. \quad (19)$$

Considering the cost function $\mathcal{G}_t$ as a Lyapunov function guarantees the stability of the agents by moving them to their associated local minima [6]. This results in the following control law

$$u_{tm} = -k(p_{tm} - C_{V_{tm}}), \quad (20)$$

where k is a positive gain and the Voronoi cells are being updated continuously over time.

Corollary 1.1: For the closed-loop system obtained by controller (20), the sets of centroidal Voronoi configurations are subspaces of Q and Q_t. If the set of centroidal Voronoi configurations on Q is finite, the teams nucleus and agents location converge to centroidal Voronoi configurations.

Proof: By applying the control law (20), the agent positions converge asymptotically to the set of critical points of $\mathcal{G}_t$ according to Proposition 3.1 in [6], $p_{tm} \to C_{V_{tm}}$. It is concluded that $l_t \to C_{V_t}$ by considering $p_{tm} \to C_{V_{tm}}$ in equation (6) with a similar argument as in the proof of Theorem 3.1. Therefore, the teams nucleus and agents locations converge to the sets of centroidal Voronoi configurations. If the set of centroidal Voronoi configurations on Q is finite, then the limit of $\mathcal{P}_t$ is unique and equals to one of the centroidal Voronoi configurations. Consequently, the teams nucleus converges to a centroidal Voronoi configuration according to Theorem 3.1. ■

Remark 1: The region assigned to each team changes over time. Since the assigned region is time-varying, there may exist robots located outside the assigned region of their team. Consequently, there might be empty Voronoi cells, which belong to the outside agents. In this case, the outside agents move towards the centroids of their Voronoi cells by using the controller (20). It has been assumed throughout this paper that there cannot be any agent without an assigned region.

Remark 2: According to Remark 1, there is a possibility that an agent can be located in a region covered by another team. While this agent is moving towards the assigned region of its team, it may collide with agents of another team. We have considered here the coverage problem without the possibility of agents colliding with each other.

IV. Modeling a Distributed Network of Agents

The agents need to be modeled with respect to their actions including sensing, communication, computation and control. The behavior of the agents in a given network is then describable as how they interact to perform the assigned task(s). In essence, the communication network and the data flow in a system of agents need to be investigated. The characteristics of the agents communication and the protocols are discussed in this section.

A. Characteristics of the agents communication

An agent in the given framework is introduced here as the m^{th} element of the t^{th} team. Each agent is capable of allocating the state of the system and performing the required operations. The agent that is located at p_{tm} can move over the space at any time for any period of time $\delta t_{tm} \in \mathbb{R}_+$ following the enforced first order dynamics (19). The agent has access to both its position p_{tm} and its associated team nucleus l_t received through the communication with the neighboring agents. Each agent also computes the control pair $(\delta t_{tm}, u_{tm})$ for the robot at p_{tm}. The agents on the borders of teams also receive the information on the nucleus of the neighboring teams. This information is provided by communicating with the neighboring agents belonging to the neighboring teams. Furthermore, it can detect the neighboring agents within the radius R_{tm} of the agent at p_{tm}. In addition to this, each agent is able to send/receive information to/from the other agents within the communication radius $R_{tm} \in \mathbb{R}_+$. It is also assumed that the agent can adjust the radius R_{tm} to ensure the minimum communication, especially in the presence of a limited communication bandwidth.

B. Team and agent's Voronoi maintenance

In order to calculate the Voronoi partition, each agent needs to know the position of the other agents in its neighborhood and also the agents on the borders require the nucleus of the neighboring teams to calculate the borders of the teams. To this end, the designed communication network should provide the motion control scheme with the required information. In this section, the Voronoi computation and maintenance is handled through a new algorithm proposed here for an asynchronous communication to maintain the data flow inside each team and among the teams.

To initialize, it is assumed that all the agents have the nucleus of their associated team and other agents position at the initial time. The agents employ an adjustable communication radius-based algorithm as presented in Algorithm 1 to receive the position information for computing the Voronoi cells. It should be noted that since the agents on the boundaries need the nuclei of the other teams to compute their Voronoi cells, Algorithm 1 guarantees a proper communication radius to receive the information from the neighboring teams while making sure that the interior agents will not be increasing their communication radius unnecessarily. To this aim, two communication radiuses are defined; first, the interior agents communicate by an adjustable radius R_{c1} to receive the data from the neighboring agents belonging to the same team. Then, if an agent detects the agents belonging to the neighboring teams, it means that the associated agent is on the boundaries, and, therefore, the communication radius is adjusted to R_{c2} to collect information on the nucleus of the neighboring teams. Finally, when the two Voronoi cells are obtained by (2) and (3), their intersection is computed as the final Voronoi cell for the agents on the boundaries.

Due to the nature of the team-based communication and considering the fact that agents need to know the nucleus of the team, the communication network is designed to transfer data throughout each team's network towards the nucleus. Then, the updated nucleus is distributed among the agents of each team. To maintain the Voronoi cells, the nucleus of the team needs to be updated. To do so, we propose Algorithm 2 based on the greedy forwarding strategy presented in [10] that sends the data packet that contains information regarding mass of the Voronoi cells and the positions of the robots towards the nucleus of the team. The information packet that is sent consists of two data strings; the first one contains the summation over the multiplication of the mass and the location of each agent and the second string represents the summation over the multiplication of the mass of the Voronoi cells that the data packet has been passed through. In other words, the algorithm is designed in a way that each agent sends information to the agent closest to the nucleus in its neighborhood. Every agent that receives the data packet sums up all the received data and forwards it towards nucleus.

Algorithm 1 Sensing algorithm for teams of robots

initialize sensing radius R_{tm}
detect all agents p_{tm} in the same team and all nucleus l_t of the neighboring team's agents within the radius R_{tm}
update $P^m(t_m)$ and $\mathcal{L}^m(t_m)$ to compute $\mathcal{Q}(p_{tm}, R_{tm})$ and $\mathcal{Q}(l_t^m, R_{tm})$
$R_{c1} = \max_{q \in \mathcal{Q}(p_{tm}, R_{tm})} \|p_{tm} - q\|$
if detect neighboring team's agents **then**
  $R_{c2} = \max_{q \in \mathcal{Q}(l_t^m, R_{tm})} \|l_t^m - q\|$
else
  $R_{c2} = 0$
end if
while $R_{tm} \leq 2 \max\{R_{c1}, R_{c2}\}$ **do**
  $R_{tm} = 2 \max\{R_{c1}, R_{c2}\}$
  detect all agents p_{tm} in the same team and all nucleus l_t of the neighboring team's agents within the radius R_{tm}
  update $P^m(t_m)$ and $\mathcal{L}^m(t_m)$
  compute $\mathcal{Q}(p_{tm}, R_{tm})$ and $\mathcal{Q}(l_t^m, R_{tm})$
  $R_{c1} = \max_{q \in \mathcal{Q}(p_{tm}, R_{tm})} \|p_{tm} - q\|$
  if detect neighboring team's agents **then**
    $R_{c2} = \max_{q \in \mathcal{Q}(l_t^m, R_{tm})} \|l_t^m - q\|$
  else
    $R_{c2} = 0$
  end if
end while
$R_{tm} = 2 \max\{R_{c1}, R_{c2}\}$
$V_{tm} = \mathcal{Q}(l_t^m, R_{tm}) \cap \mathcal{Q}(p_{tm}, R_{tm})$

Every node in the network performs the forwarding process until the time based stopping criterion is satisfied. This criterion is defined by multiplying the number of agents in each team by the maximum operating period of time $\max\{\delta t_{t1}, \ldots, \delta t_{tn_t}\}$ for t^{th} team. The agent closest to the nucleus sums up all the received data and calculates the new nucleus of the team. As soon as the agent closest to the nucleus calculates the updated nucleus, it disseminates the data packet over the team network by the communication radius that has been already calculated in the Voronoi computation. Each agent that receives the new nucleus also runs the same data transfer algorithm until the data packet reaches the borders. To realize whether agents are sharing boundaries with other teams, each agent can run a quick check to see if any of its boundaries has been drawn by using the teams nucleus or not. This is considered as the stopping criterion for the message routing. Now, the agents have been informed about the new nucleus of the team, and hence, the ones on the border communicate with the agents belonging to the other teams that they share boundaries with to get their team nucleus to draw their Voronoi cells. The algorithm for drawing the associated Voronoi regions of the agents that might require communication with other teams is given in Algorithm 1. Now that the team boundary has been updated, the agents will need to repartition their associated space.

Algorithm 2 The greedy forwarding algorithm for nucleus calculation

Require: given $\delta t_{t1}, \ldots, \delta t_{tn_t}$
agent m detects agent m' as the neighbor closed agent towards the nucleus l_t
if $m' \neq m$ **then**
  send the pair $(M_{V_{tm}} p_{tm}, M_{V_{tm}})$ to the agent m'
else$(\alpha_t, \beta_t) = (M_{V_{tm}} p_{tm}, M_{V_{tm}})$
end if
$T_m = \delta t_{tm}$
while $T_m < n_t \max\{\delta t_{t1}, \ldots, \delta t_{tn_t}\}$ **do**
  $T_m \leftarrow T_m + \delta t_{tm}$
  if receives data packet **then**
    sum up all the received data
    if $m' \neq m$ **then**
      send the summation to the agent m'
    end if
    if $m' = m$ **then**
      $(\alpha_t, \beta_t) \leftarrow (\alpha_t, \beta_t)+$ the summation of all the received data packets
    end if
  end if
end while
if $m' = m$ **then**
  compute the update nucleus by (6) as $l_t = \alpha_t / \beta_t$.
end if

V. Simulation Results and Discussion

In this section, the presented team based partitioning is illustrated through a simulation examples. 50 agents are deployed on a 5×10 rectangular in five teams with equal number of agents. To illustrate the case that there are multiple important regions, a Gaussian functions with 3 picks are chosen. The density function is chosen as the following Gaussian function

$$\varphi(x, y) = \sum_{i=1}^{n_p} \exp\left(-\frac{(x - a_i)^2 + (y - b_i)^2}{2\sigma^2}\right), \qquad (21)$$

where n_p is the number of centers, (a_i, b_i) represents the coordinate of the centers and σ is the variance.

Example

The first density function is chosen as a Gaussian function given by (21) where $n_p = 3$ and $\sigma = 0.25$. Figure 1 illustrates the density function over the given space. A

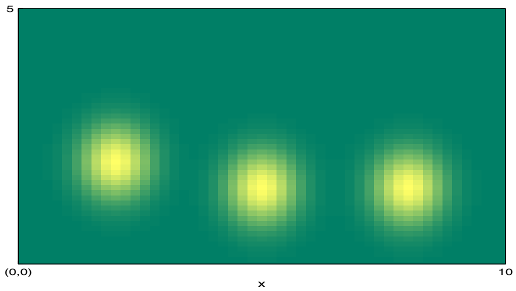
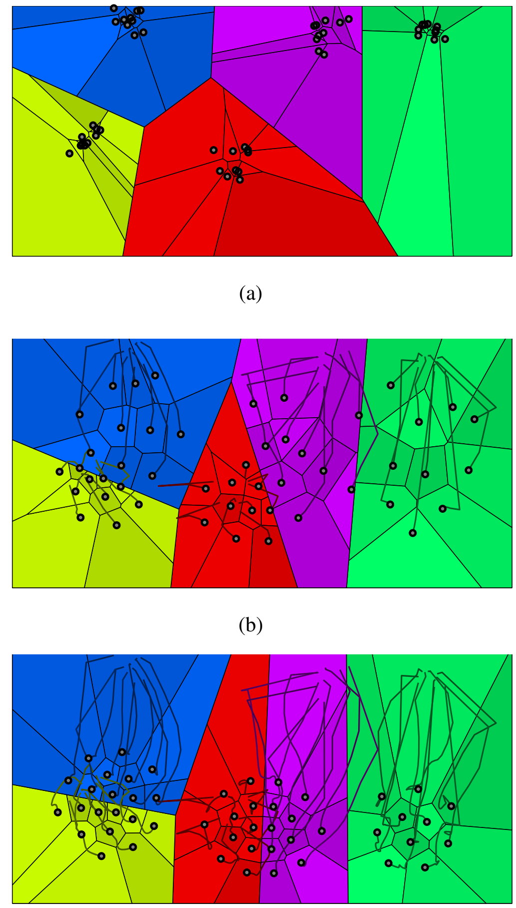


Fig. 1. Gaussian density function with the centers chosen at $(2, 2)$, $(5, 1.5)$ and $(8, 1.5)$.

random configuration is chosen as the initial configuration of the nucleus of the teams. Then, the robots are deployed in the region associated with each team. As the proposed control law moves the agents to the centroid of their Voronoi cells, the teams nucleus also moves towards their centroid. Figure 2 shows the solution to the coverage problem. It can be observed that along with partitioning in the team level, the agents also divide the region assigned to each team considering the given density function. Also, due to the interdependency of the teams and their interior agents, the teams spread over the given space to provide the optimum coverage while their interior agents also partition each team's Voronoi cell into smaller partitions considering the main density function. The trend shows that the teams also tend to concentrate on the region given by the density function, and, hence, the team partitioning is done collaboratively. As observed from Figure 2c, the teams in the left and the middle divide the region in a way that they optimize their associated sensing function according to the defined cost function both in the team-level (7) and agents-level (8) to ensure the best coverage. The traversed path by each agent to reach to the optimum configuration is shown in Figure 2c.



(c)

Fig. 2. The change from the initial to the final configuration is shown from (a) to (c) for Example 1.

VI. Concluding Remarks

To adapt with the complexity of the coverage problems in real-world applications, a team-based coverage approach is presented in this paper. The proposed approach solves the problem in a more local way leading to the partitioning of the main region into multiple subregions associated with multiple deployed teams. This is beneficial to handle the tasks that require a more diverse group of robots due to their complexity. This is to say that due to the diversity of the coverage problems, the need for deploying robots with different dynamics or communication characteristics might be inevitable, and, hence, the proposed team-based approach is an attempt to modify and improve existing methods in a way that the teams of robots can divide the region among themselves based on their capabilities. This provides the means for autonomous deployment of multiple teams of robots when there is a need for different types of robots form both communication and dynamics perspective.

References

[1] A. Wallar, E. Plaku, and D. Sofge, "Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 969–980, July 2015.

[2] A. Macwan, G. Nejat, and B. Benhabib, "Optimal deployment of robotic teams for autonomous wilderness search and rescue," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4544–4549.

[3] S. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, April 2015.

[4] O. Baron, O. Berman, D. Krass, and Q. Wang, "The equitable location problem on the plane," *European Journal of Operational Research*, vol. 183, no. 2, pp. 578–590, 2007.

[5] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. New York, NY, USA: John Wiley & Sons, Inc., 1992.

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

[7] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.

[8] F. Sharifi, A. Chamseddine, H. Mahboubi, Y. Zhang, and A. Aghdam, "A distributed deployment strategy for a network of cooperative autonomous vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 737–745, March 2015.

[9] M. Pavone, E. Frazzoli, and F. Bullo, "Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1259–1274, June 2011.

[10] J. Chen, Y. Guan, and U. Pooch, "Customizing GPSR for wireless sensor networks," in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Oct. 2004, pp. 549–551.