Heterogeneous Coverage and Multi-Resource Allocation in Supply-Constrained Teams

Mela Coffey and Alyssa Pierson

***Abstract*— We consider a team of heterogeneous robots, each equipped with various types and quantities of resources, and tasked with supplying these resources to multiple areas of demand. We propose a Voronoi-based coverage control approach to deploy robots to areas of demand by defining a position- and time-varying density function to represent the quality at which demand is being met in the environment. This approach allows robots to prioritize the various demand locations in a continuous, distributed fashion. We present analyses to show that our controls drive the robots to critical points in the environment, along with simulations and hardware-in-the-loop experiments to demonstrate our approach.**

I. INTRODUCTION

Control of multi-robot systems has become an active research topic, and rightly so; in most scenarios, a team of multiple robots can accomplish much more than a single one. A popular application of multi-robot systems is the delivery of goods to humans. For example, in an emergency situation in which medical supplies are needed in multiple locations for first responders to treat victims, a fleet of medical drones would be useful to deploy medical resources to those locations. In such situations where terrain and environmental conditions may be dangerous, the use of drones ensures human safety. The problem then becomes how to deploy the robots to meet the demand of medical supplies given the resources each robot carries, i.e. the capacity of each robot.

To address such a resource allocation problem, we propose a Voronoi-based coverage control approach. Coverage control deploys robots to areas of importance in an environment to effectively "cover" those areas. In Voronoi-based coverage control, the space is partitioned into Voronoi cells, where each robot is responsible for covering the space in its cell, and the importance is represented by a density function $\phi(q)$, where q are points in the environment. Voronoi-based coverage control poses many advantages in multi-robot systems, including a continuous control policy, collision avoidance between teammates, adaptivity, and opportunities for distributed and decentralized implementations. Distributed coverage control was first proposed in [1] and has since been utilized and extended for many coverage control applications. For example, the authors of [2] proposed adaptive (learning) density functions. In [3], the authors proposed adaptive weightings of weighted Voronoi cells.

Mela Coffey and Alyssa Pierson are with the Department of Mechanical Engineering at Boston University, Boston, MA 02115, USA. mcoffey@bu.edu, pierson@bu.edu

This work supported in part by a Boston University startup grant. We are grateful for their support.

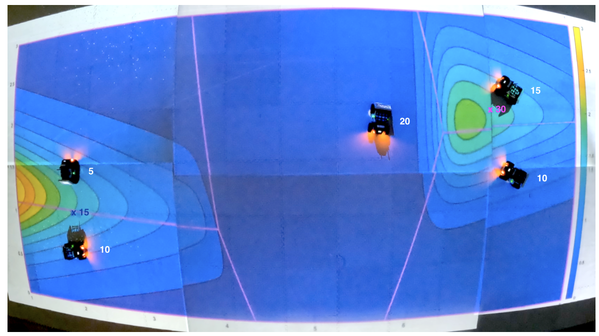


Fig. 1. Final configuration of a team of 5 robots supplying multiple resources of a single type to 2 different locations (×). Quantities of resources carried/demanded are printed to the right of the robot/×. The colorbar indicates density in the space on a scale of 0 to 3, and lines represent Voronoi boundaries.

In this paper, we utilize Voronoi-based coverage control to deploy supplying robots to areas of demand. We consider a heterogeneous team of robots, each equipped with various types of resources and varying capacities for those resources. The robots are tasked with supplying their resources to multiple areas of demand, each requiring different types and quantities of resources. We propose a position-varying, time-varying density function which takes into account the capacities of each robot, the demand, and the positions of the robots with respect to the demand. We propose a controller to minimize the locational cost function for our heterogeneous team. We show that our controller drives the robots to minimize the locational cost. The main contributions of this paper are:

- Coverage control for deploying a heterogeneous team with supply constraints to multiple demand locations;
- Analyses to prove minimization of locational cost; and
- Simulations and hardware-in-the-loop experiments to demonstrate performance.

Related Work

A common approach to representing heterogeneous robots within Voronoi-based methods invokes weighted Voronoi diagrams [3], [4], [5], [6], [7], [8], [9], [10], [11], where each robot's Voronoi cell is weighted by a constant or variable. Intuitively, a lower relative weight yields a smaller cell relative to its neighbors. To address a supply-and-demand problem, one might weigh robot i's Voronoi cell by its supply capacity. While this method ensures that robots with larger capacities cover more demand than its neighbors, it does not account for multiple resource types among the robots.

Recent coverage control work related to task allocation focuses on deploying a team of heterogeneous robots to different locations. For example, [12], [13], [14] assign robots with different sensing capabilities to regions requiring that sensor. One can think of these approaches as deploying robots with different resource types to areas requiring those resources types. In [11] and [15], robots with differing dynamical abilities are deployed to multiple areas of importance. The authors of [16] assign a fidelity level and sensing function to each sensing type before deployment. In [17], regularized optimization methods are implemented to adapt to failed agents at various event types.

Other approaches to resource allocation focus on optimization methods, routing, and scheduling to allocate robots to demand locations. The authors of [18] propose a dynamic programming algorithm for multi-robot task allocations. In [19] and [20], authors focus on routing and task allocation for robot teams in smart warehouses. In [21], authors propose a scheduling algorithm for charging mobile robots. The authors of [22] propose methods for multi-agent task allocation based on capabilities and features of each robot.

From the perspective of resource allocation using robot teams, prior work in heterogeneous coverage control has focused on either deploying resource types to multiple locations, or deploying a single resource of varying capacity to a single location. To our knowledge, the resource allocation problem considering multiple resource types with varying capacities remains an open problem. While it is indeed possible to compute a globally optimal robot assignment, then deploy robots to demand using Voronoi-based methods, such an approach does not allow for a continuous, distributed control policy. In this paper, we propose a Voronoi-based coverage control method capable of distributed implementations to deploy a heterogeneous robot team to multiple demand locations, where each robot has a different capacity of each resource type, and each demand location requires different resource types and quantities.

The remainder of this paper is organized as follows: Section II provides an introduction to heterogeneous coverage control. Section III details our proposed density function, cost function, controller, and includes analyses to prove performance. In Section IV, we provide simulations, comparing our proposed controller against that which does not consider capacity constraints. Section V contains details and results from hardware-in-the-loop experiments. We state our conclusions and future work in Section VI and provide detailed computational results in Section VII.

II. Preliminaries and Background

Consider a team of N robots in a bounded, convex environment $Q \subset \mathbb{R}^2$, with points in Q denoted q, and robot positions denoted $p_i \in Q$, $i \in \mathcal{R}$, $i = 1, \dots, N$. In this section, we provide a brief background on classic Voronoi-based coverage control, and introduce the recent work on coverage control for heterogeneous multi-robot teams.

Traditional Voronoi-based coverage control [1] partitions the environment Q into N Voronoi cells, and deploys robots according to a locational cost function using Lloyd's algorithm [23], [1], with the goal of covering the environment Q according to the density function $\phi : \mathbb{R}^2 \to \mathbb{R}$. The density function represents areas of importance; a greater value of $\phi(q)$ indicates greater importance in coverage at point q.

To address a heterogeneous robot sensing team, i.e. sensing capabilities that vary among agents, the authors of [13], [14] create one Voronoi partition for every sensing type j, $j \in \mathcal{D}$, $j = 1, \dots, M$. Each partition is assigned a different density function $\phi_j(q)$ representing the importance associated with sensor j. Robots equipped with sensor j are considered in the partition $\mathcal{P}^j = \{i \mid j \in T(i)\}$, where $T(i)$ is the set of sensors carried by robot i. The environment Q is then divided into M Voronoi partitions [13], [14] by

$$V_i^j = \{q \in Q \mid \|p_i - q\| \leq \|p_l - q\|, \forall l \in \mathcal{P}^j, i \neq l\}. \quad (1)$$

Robots are then deployed using Lloyd's algorithm to minimize the cost function [13], [14]

$$\mathcal{H}(p_1, \dots, p_n) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{D}} \int_{V_i^j} f(\|q - p_i\|) \phi_j(q) dq, \quad (2)$$

where $f : \mathbb{R} \to \mathbb{R}$. The mass and centroid [13], [14] for robot i's Voronoi cell in partition j are, respectively,

$$\begin{aligned} m_i^j &= \int_{V_i^j} \phi_j(q) dq \\ c_i^j &= \frac{\int_{V_i^j} q\phi_j(q)dq}{\int_{V_i^j} \phi_j(q)dq} = \frac{1}{m_i^j} \int_{V_i^j} q\phi_j(q)dq, \end{aligned} \quad (3)$$

and the robots adhere to the control law

$$\dot{p}_i = u_i, \quad (4)$$

where $u_i = -\kappa \sum_{j \in T(i)} \int_{V_i^j(p)} \frac{\partial f(\|q - p_i\|)}{\partial p_i} \phi_j(q) dq$, and κ is a positive proportional gain constant [13].

Note that with the heterogeneous partitioning (1), collision avoidance is not guaranteed for those robots that do not share the same partition. For example, if robot 1 only has sensor type 1, and robot 2 only has sensor type 2, then those robots will not "see" one another, because they cannot share a Voronoi boundary, and there is a chance for collision. However, in sensor coverage, robots cover a large space, and at such a large scale, collision avoidance can be locally managed. In the following section, we describe our proposed implementation to accomplish a resource allocation task, considering not only resource types, but also resource quantities.

III. A Coverage Control Approach to Resource Allocation

In this paper, we consider robots that each carry different types and quantities of resources. We use Voronoi-based coverage control to deploy a heterogeneous team of robots to various locations in need of resources.

A. Partitioning the Space

Each robot i carries (supplies) $s_i^k \in \mathbb{R}_+$ resources of type $k \in \mathcal{S}$. We wish to deploy robots to M locations $\mu_j \in Q$, $j \in \mathcal{D}$, $j = 1, \ldots, M$, which require (demand) $d_j^k \in \mathbb{R}_+$ resources. Given multiple demand locations, and that each robot has a different ability to meet that demand, we define M different Voronoi partitions in the environment Q. We assign robot i to partition j if robot i can supply resources to demand j. In other words, if demand j requires resource type k, we assign all robots that carry resource type k to partition j. More formally, we define the set of robots assigned to partition j as

$$\mathcal{P}^j = \{i \mid \exists k : d_j^k > 0 \wedge s_i^k > 0\}. \quad (5)$$

Then, the Voronoi cell for robot $i \in \mathcal{P}^j$ is given by

$$V_i^j = \{q \in Q \mid \|q - p_i\| \leq \|q - p_l\|, \forall l \in \mathcal{P}^j, \forall l \neq i\}.$$

B. Defining the Density

Recall that the density function ϕ represents areas of importance in the environment Q. Thus, in a supply-and-demand scenario, one can think of ϕ as the relative demand of resources at any point $q \in Q$. We propose the following time-varying and position-varying density function to deploy robots based on demand and robot capacities:

$$\phi_j(q, p, t) = \frac{D_j}{\sum_{l \in \mathcal{P}^j} S_l}, \quad (6)$$

where

$$D_j = \sum_{k \in \mathcal{S}} d_j^k \exp\left[-\frac{1}{\sigma}(q - \mu_j)^\intercal \Sigma^{-1}(q - \mu_j)\right], \quad (7)$$

$$S_l = \sum_{k \in \mathcal{S}, d_j^k \neq 0} s_l^k \exp\left[-\frac{1}{\sigma}(q - p_l)^\intercal \Sigma^{-1}(q - p_l)\right], \quad (8)$$

σ is a positive constant, and Σ is a constant covariance matrix. Note that, although we sum over all resource types in (7) and (8), the partition assignment (5) helps distinguish between resource types. While traditional coverage control utilizes density functions that depend on q only, our proposed density function, for supply-and-demand applications, depends additionally on robot positions $p(t)$, and thus time t. Note, however, that classic and heterogeneous Voronoi-based coverage control do not utilize time-varying density functions. In fact, we can only use Lloyd's algorithm with quasi-static time-varying density functions [24]. Therefore, we need to implement control laws compatible with time-varying density functions.

C. Serving Heterogeneous Multi-Resource Demand

In order to deploy robots according to the relative importance (6), we require a method to compute the control input of time-varying density functions. The works [25], [26], [27] propose a solution for converging time-varying density functions, but requires slowly varying density functions and estimated solutions. Because our proposed density function (6) relies on $p(t)$, we require a control law suited for more rapidly varying density functions with more precise solutions. We therefore implement a variation of the minimum-energy, constraint-driven approach for coverage control with time-varying density functions [28].

For this paper, we let $f(\|q - p_i\|) = \|q - p_i\|^2$ in (2), and define the locational cost $J(p, t)$ as the sum of individual costs J_i^j of each robot $i \in \mathcal{R}$ in each of their assigned partitions j. Formally, $J(p, t) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{D}} J_i^j(p_i, t)$. We define the individual cost of robot i in partition j as $J_i^j(p, t) = \frac{1}{2} m_i^j \|p_i - c_i^j\|^2$, where m_i^j and c_i^j are given by (3). We can then write the total cost as

$$J(p, t) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{D}} \frac{1}{2} m_i^j \|p_i - c_i^j\|^2. \quad (9)$$

We let each robot i obey the single integrator dynamics (4), where u_i is computed by solving the optimization problem

$$\begin{aligned} \min_{u_i} \quad & \|u_i\|^2 \\ \text{s.t.} \quad & \sum_{j \in \mathcal{D}} (-m_i^j (p_i - c_i^j)^\intercal (I - \frac{\partial c_i^j}{\partial p_i}) u_i \\ & - \frac{1}{2} \|p_i - c_i^j\|^2 \frac{\partial m_i^j}{\partial p_i} u_i) \geq \sum_{j \in \mathcal{D}} (-\alpha(-J_i^j(p, t)) \\ & - m_i^j (p_i - c_i^j)^\intercal \frac{\partial c_i^j}{\partial t} + \frac{1}{2} \frac{\partial m_i^j}{\partial t} \|p_i - c_i^j\|^2), \end{aligned} \quad (10)$$

where I is the identity matrix, and α is an extended class $\mathcal{K}$ function, $\alpha : p \in \mathbb{R} \mapsto \alpha(p) \in \mathbb{R}$, superadditive for $p < 0$, i.e. $\alpha(p_1 + p_2) = \alpha(p_1) + \alpha(p_2)$, $\forall p_1, p_2 < 0$. For the partial derivatives of c_i^j and m_i^j, see Appendix (Section VII). Remarkably, the control input computed by (10), minimizes the cost (9) in a decentralized fashion given the dynamics (4), as formalized in Lemma 1.

Lemma 1: Let the time-varying cost function $J(p, t)$ be defined by (9). Then, a group of N robots obeying the single-integrator dynamics (4) minimizes the cost J in a decentralized fashion if each robot executes the control input solution of the optimization problem (10).

Proof: We begin by showing that the formulation of our cost function satisfies the set of conditions outlined in [29]. A general expression for the locational cost that leads to decentralized control laws is

$$J(p, t) = \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{D}} \sum_{l \in \mathcal{N}_i^j} J_{il}^j(\|p_i - p_l\|, t), \quad (11)$$

where $\mathcal{N}_i^j$ is the neighbor set of robot i in partition j (i.e. the set of all robots with which robot i shares a Voronoi boundary in partition j), and J_{il}^j is the pairwise performance cost between neighbors i and l in partition j. We require $J_{il}^j : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$, $J_{il}^j(\|p_i - p_l\|, t) = J_{li}^j(\|p_l - p_i\|, t)$ symmetric, $J_{il}^j \geq 0, \forall p \in \mathbb{R}^n, \forall t \in \mathbb{R}_+$, such that $J(p, t) \geq 0$ $\forall p \in \mathbb{R}^n$, $\forall t \in \mathbb{R}_+$ [29]. Given the formulation of our cost function in (9), and that $m_i^j \geq 0$ $\forall p \in \mathbb{R}^n$, $\forall t \in \mathbb{R}_+$, then $J(p, t) \geq 0$ $\forall p \in \mathbb{R}^n, \forall t \in \mathbb{R}_+$. Additionally, given the graph topology induced by the Voronoi partition j, then (9) can be written in the form (11). Thus, our cost functions satisfies the conditions outlined in [29]. Therefore, by [30], a group of N robots obeying single integrator dynamics minimizes J in a decentralized fashion if each robot executes the control input solution of the optimization problem [30]

$$\begin{aligned} \min_{u_i,\delta_i} \quad & \|u_i\|^2 + |\delta_i| \\ \text{s.t.} \quad & -\frac{\partial J_i}{\partial p_i}u_i \geq -\alpha(-J_i(p)) + \frac{\partial J_i}{\partial t} - \delta_i. \end{aligned} \tag{12}$$

We can reformulate (12) for our cost J by taking the partial derivatives of (9) with respect to p_i and t as follows:

$$\begin{aligned} \frac{\partial J_i^j}{\partial p_i} &= \sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{D}} -m_i^j(p_i - c_i^j)^\mathsf{T}\frac{\partial c_i^j}{\partial p_i} + \frac{1}{2}\frac{\partial m_i^j}{\partial p_i}\|p_i - c_i^j\|^2, \\ \frac{\partial J_i^j}{\partial t} &= \sum_{i\in\mathcal{R}}\sum_{j\in\mathcal{D}} \frac{1}{2}\frac{\partial m_i^j}{\partial t}\|p_i - c_i^j\|^2 + m_i^j(p_i - c_i^j)^\mathsf{T}(\dot{p}_i - \frac{\partial c_i^j}{\partial t}). \end{aligned} \tag{13}$$

Then, plugging in (13) and (4), we can reformulate (12) to exclude the slack variables, and obtain (10). Therefore, by [30], our control policy minimizes the cost J, which completes the proof. ■

Lemma 1 shows that the control input (10) minimizes the cost (9). We can now show that with input (4) computed from the optimization problem (10), each robot i converges toward a critical point, which is that at which the cost $J_i(p_i, t)$ is minimized.

Proposition 1: Let robot i with planar position p_i be assigned to at least one partition j and evolve according to the control law (4), where u_i is attained by solving the optimization problem (10). Let $p_i = p_{c,i}$ be the location at which robot i is located at a critical point of its cost $J_i(p_i, t)$. Then, the distance between p_i and $p_{c,i}$, $\|p_i - p_{c,i}\|$, is uniformly bounded.

Proof: First note that, at the critical point $p_i = p_{c,i}$, $p_i - p_{c,i} = 0$, and we wish to prove that $\|p_i - p_{c,i}\|$ is uniformly bounded about $p_i - p_{c,i} = 0$. Since $J_i(p_i, t)$ is continuously differentiable, strictly increasing, and strictly positive when robot i is assigned to at least one partition j, then there exist class $\mathcal{K}$ functions $\alpha_{1,i}(p_i)$ and $\alpha_{2,i}(p_i)$ such that

$$\alpha_{1,i}(\|p_i - p_{c,i}\|) \leq J_i(p_i, t) \leq \alpha_{2,i}(\|p_i - p_{c,i}\|). \tag{14}$$

Now, we compute the time derivative of the associated with robot i to be

$$\dot{J}_i(p_i, t) = \sum_{j\in\mathcal{D}} \dot{J}_i^j(p_i, t) = \sum_{j\in\mathcal{D}} \frac{\partial J_i^j}{\partial p_i}\dot{p}_i + \frac{\partial J_i^j}{\partial t}.$$

Plugging in (9) and (13), we obtain

$$\begin{aligned} \dot{J}_i(p_i, t) = \sum_{j\in\mathcal{D}} &[m_i^j(p_i - c_i^j)^\mathsf{T}(I - \frac{\partial c_i^j}{\partial p_i})u_i \\ &+ \frac{1}{2}\|p_i - c_i^j\|^2\frac{\partial m_i^j}{\partial p_i}u_i \\ &- m_i^j(p_i - c_i^j)^\mathsf{T}\frac{\partial c_i^j}{\partial t} + \frac{1}{2}\frac{\partial m_i^j}{\partial t}\|p_i - c_i^j\|^2]. \end{aligned}$$

Summing over each demand $j \in \mathcal{D}$ in (10), and given the superadditive property of $\alpha(\cdot)$, we obtain

$$\dot{J}_i(p_i, t) \leq \alpha(-J_i(p_i, t)). \tag{15}$$

Let $\bar{\alpha} = -\alpha(-r)$, then $\bar{\alpha}$ is also an extended class $\mathcal{K}$ function, and $\alpha(-J_i(p_i, t)) = -\bar{\alpha}(J_i(p_i, t))$. Thus, (15) becomes

$$\dot{J}_i(p_i, t) \leq -\bar{\alpha}(J_i(p_i, t)) \quad \forall\|p_i - p_{c,i}\| \geq \mu_i > 0. \tag{16}$$

We can now use Theorem 4.18 in [31] to complete the proof. Let $r_i > 0$ such that $B_{r_i} \subset Q$, and suppose that $\mu_i < \alpha_{2,i}^{-1}(\alpha_{1,i}(r_i))$. Then, by Theorem 4.18 in [31], for all initial states $p_{0,i}$ satisfying $\|p_{0,i} - p_{c,i}\| \leq \alpha_{2,i}^{-1}(\alpha_{1,i}(r_i))$, and since (14) and (16) are satisfied, then $\|p_i - p_{c,i}\|$ is uniformly bounded, which completes the proof. ■

Corollary 1: When the cost of each robot $J_i(p_i, t)$ is bounded, then the total cost of the team $J(p, t)$ is bounded.

Proof: The total cost $J(p, t)$ is given by (9). Given that the costs associated with each of the N robots is bounded, then, summing over all robots $i \in \mathcal{R}$, the total cost $J(p, t)$ is also bounded. ■

Remark 1: By Lemma 1, our control input (10) minimizes the cost (9). Therefore, the critical points to which the robots converge will be either a local minimum or a saddle point within each robot's cell. In practice, we notice that each robot converges to one of the centroids within its cell, as shown in our simulations and experiments, and each robot oscillates around the centroid to which it converges, as can be seen in the supplemental video.

Remark 2: Such oscillating behavior is a feature of the control input (10), whose constraint serves to drive J_i to 0. However, since $J_i \neq 0$ when robots belong to more than one partition, $u_i \neq 0$.

Proposition 1 states that, under our proposed control policy, each of the N robots will drive toward a critical point of the cost (9). Furthermore, those critical points cannot be local maxima, as formalized by Remark 1. In the next section, we present simulations to demonstrate the performance of our proposed control policy.

IV. SIMULATIONS

We performed simulations to evaluate the ability of our control policy to reduce the cost (9). We consider two scenarios: 1) the initial demand parameters M, μ_j, and d_j^k remain constant, while the initial team parameters N, p_i, and s_i^k are randomized for each trial, and 2) the initial team parameters remain constant, while demand varies. We performed 50 trials of each scenario for a total of 100 trials, the initial conditions for which are summarized in Table I.

Additionally, we present the same 100 trials alongside a baseline simulation, in which robots are assigned to partitions based on whether or not they have a resource type k, as in our proposed approach, but robot capacities are not considered. To achieve this, we "hide" the capacities by setting $s_i^k = 1$ for all $s_i^k \neq 0$, and $s_i^k = 0$ otherwise. Similarly, we set $d_j^k = 1$ for all $d_j^k \neq 0$, and $d_j^k = 0$ otherwise. These values are used to compute the density function and ultimately the

TABLE I
SIMULATION CONDITIONS

Scenario	Constants	Randomized Variables
1 Randomized Team	$k = \{1, 2\}$ $\mu_1 = (-0.5, -0.5)$ $\mu_2 = (0.5, 0.5)$ $d_1^k = \{75, 0\}$,$d_2^k = \{20, 60\}$	$N \in [5, 15]$ $s_i^k \in [0, 20] \quad \forall i, k$ $p_i \in Q \quad \forall i$
2 Randomized Demand	$k = \{1, 2\}$, $N = 10$ $p_{i,x_0} = \{-.9, -.5, .5, .9,$ $.9, .9, .5, -.5, -.9, -.9\}$ $p_{i,y_0} = \{-.9, -.9, -.9,$ $-.9, 0, .9, .9, .9, .9, 0\}$ $s_i^1 = \{1, 0, 5, 3, 15,$ $20, 16, 18, 1, 1\}$ $s_i^2 = \{2, 4, 0, 1, 1,$ $1, 1, 1, 15, 20\}$	$M \in [2, 4]$ $p_i \in Q$ $d_i^k \in [0, 50] \quad \forall j, k$

control input. Performing this binary version of coverage control deploys robots based solely on whether they have resource type k, rather than the capacity of the robots. We refer to this case as the Hidden Capacity case. In order to compare this baseline with our proposed algorithm, we compute the total cost with the true capacities. In other words, although the actual capacities are hidden from the controller, we compute the total cost as if the capacities exist, so as to provide a comparison for our approach.

We performed all simulations in MATLAB and implemented the robot dynamics (4) with u_i computed from (10), limiting $\dot{p}_i$ to 0.1 m/s. We set $\sigma = 0.12$, Σ as the identity matrix, and $\alpha(x) = x^{\frac{1}{3}}$. Results for the 50 trials each of Scenarios 1 and 2, with both known and hidden capacities, are shown in Figure 2. Here, we present box-and-whisker plots of the total cost at the end of each trial. In both scenarios, the final cost is lowest when capacities are known, which demonstrates the ability of our control policy to deploy a heterogeneous team of supply robots to meet demand. We notice that Scenario 2 generally results in a higher final cost than Scenario 1, because robots have far fewer resources than can meet the total demand; thus, the total performance is poorer, and the cost is higher.

We also present two example trials in Figure 3, one from each scenario under the Known Capacity and Hidden Capacity cases. We provide snapshots of the final configurations, along with the cost-time plots. In the final configurations of both scenarios, all robots have converged to one of the centroids in their cell. As we can see in the animations, and as mentioned in Remark 1, once a robot reaches its centroid, it experiences small, bounded oscillations about that centroid, which is a feature of the optimization problem (10) [28]. When capacities are hidden, we observe from Figure 3 that the final configuration results in a higher density over the space compared to the known capacity case, indicating that the team performance in meeting the demand is poorer when capacities are unknown. This higher density ultimately results in a higher cost.

V. EXPERIMENTS

In addition to simulations, we performed hardware-in-the-loop experiments to demonstrate that our proposed control policy maintains performance on real robots and in real time. We implemented our control policy on a fleet of five Agilex LIMO[1] omnidirectional robots, powered by the NVIDIA Jetson Nano[2], using Robot Operating System (ROS). A desktop computer (8-core, 32GB RAM, Windows 10) sent the following linear and angular velocity steering commands [26], respectively, to each robot via Wi-Fi:

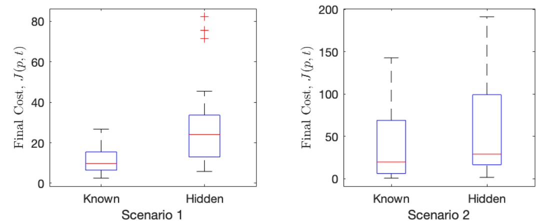


Fig. 2. Box-and-whisker plots comparing the final cost of the Known capacity and Hidden capacity cases for both scenarios. Each box results from 50 trials.

$$v_i = k_v \begin{bmatrix} \cos\theta_i & \sin\theta_i \end{bmatrix} \dot{p}_i,$$
$$\omega_i = k_\omega \arctan\left(\frac{\begin{bmatrix} -\sin\theta_i & \cos\theta_i \end{bmatrix} \dot{p}_i}{\begin{bmatrix} \cos\theta_i & \sin\theta_i \end{bmatrix} \dot{p}_i} \right),$$

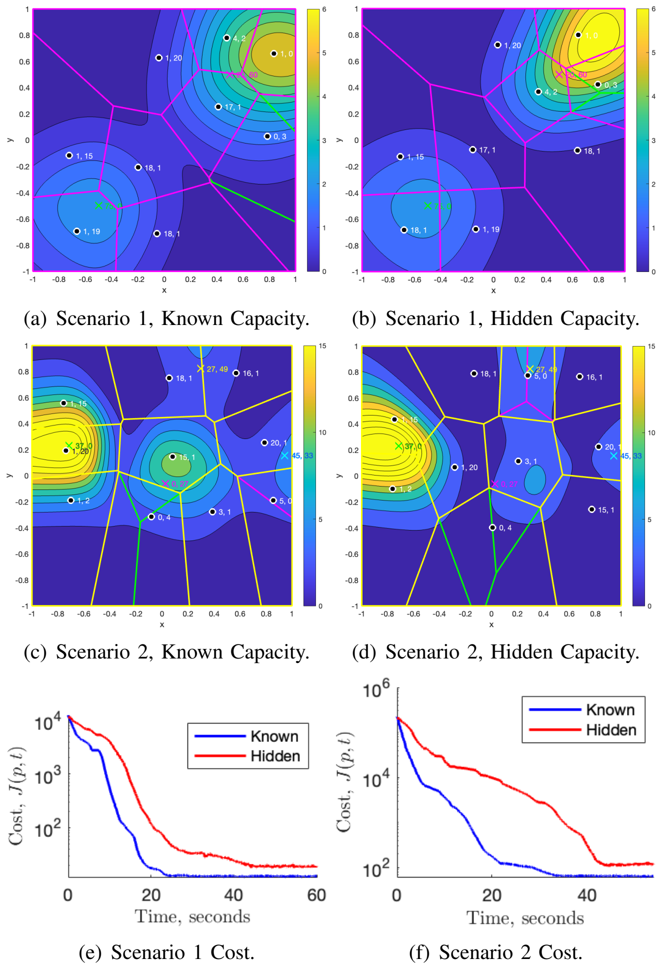
where θ_i is the heading angle with respect to the global frame, and k_v and k_ω are positive constants. The desktop computer received position and orientation information from an Optitrack Motion Capture system[3].

We present two experiments in Figures 1 and 4. For both experiments, five LIMO robots started in the same initial positions in the center of the environment, as shown in the supplemental video, and had capacities $s_i^1 = \{15, 10, 20, 5, 10\}$. In the first experiment, $\mu_1 = (2.0, 1.0)$ with $d_1^1 = 30$, and $\mu_2 = (7.0, 2.0)$ with $d_2^1 = 15$. In the second experiments, μ_1 and μ_2 remained the same, and we swapped the demand such that $d_1^1 = 15$ and $d_2^1 = 30$. We present both of these experiments in the supplemental video. Under our proposed controller, the robot with the highest demand $i = 3$ converged to the centroid associated with the highest demand in both experiments, ensuring that the robots meet the demand.

VI. CONCLUSIONS

In this work, we present a method to allocate a heterogeneous multi-robot team equipped with various resource types and quantities to various demand. We utilize Voronoi-based coverage control with a time-varying and position-varying density function, such that the density, or performance, changes as robots cover the space. Our approach serves as a method of deploying robots to multiple locations while considering resources types and quantities. We provide a continuous control policy with opportunities for distributed and decentralized implementations, and we show that it drives robots to minimize the locational cost. We present simulations comparing our work with those which do not

[1] https://global.agilex.ai/products/limo
[2] https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[3] https://optitrack.com/

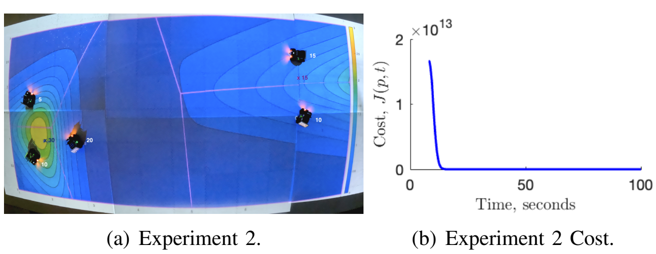


(a) Scenario 1, Known Capacity. (b) Scenario 1, Hidden Capacity.

(c) Scenario 2, Known Capacity. (d) Scenario 2, Hidden Capacity.

(e) Scenario 1 Cost. (f) Scenario 2 Cost.

Fig. 3. Example simulations for Scenarios 1 (a)-(b) and 2 (c)-(d) with Known and Hidden capacities. Here we present the final configurations of the robots and the corresponding cost-time plots. Robots are represented ○'s, and demand locations are represented by ×'s and color coded. The quantities s_i^k and d_j^k are printed next to each p_i and μ_j, respectively, for each k. The color bar to the right of each environment represents the density, and the solid lines represent Voronoi boundaries. In (a) and (b), we only see a single Voronoi partition because all robots are assigned to both partitions; thus, the two Voronoi partitions overlap in these cases. Total cost is plotted in (e) and (f).



(a) Experiment 2. (b) Experiment 2 Cost.

Fig. 4. Hardware-in-the-loop experiments with a team of five omni-directional robots. (a) Final configuration for the experiment, and (b) corresponding cost over time. Similarly to the simulations, we only see one Voronoi partition here because the two Voronoi partitions overlap, since all robots have resource type k.

consider supply constraints, and demonstrate our approach in hardware-in-the-loop experiments. Our simulation results show that our control policy reduces our locational cost and results in a lower cost than when capacity is not considered. Both our simulations and experiments show that our approach deploys robots to multiple demand locations while accounting for resource constraints.

VII. Appendix

Here, we provide additional detail in the derivation of m_i^j, c_i^j, and ϕ_j. We derive the partial derivatives of m_i^j and c_i^j using Reynolds Transport Theorem [32]. Differentiating c_i^j from (3) with respect to robot position and time using Reynolds Transport Theorem, we obtain

$$\frac{\partial c_i^j}{\partial p_i} = \sum_{l\in\mathcal{N}_i^j}\int_{V_i^j(p)} \frac{(q-p_i)\phi_j(q,p,t)(p_l-p_i)^\intercal}{m_i^j\|p_l-p_i\|^2}dq + \frac{1}{m_i^j}\int_{V_i^j(p)}(q-c_i^j)\frac{\partial\phi_j(q,p,t)}{\partial p_i}dq + \sum_{l\in\mathcal{N}_i^j}\int_{\partial V_i^j(p)}\frac{(q-c_i^j)\phi_j(q,p,t)(q-p_i)^\intercal}{m_i^j\|p_l-p_i\|}dq,$$

and

$$\frac{\partial c_i^j}{\partial t} = \frac{1}{m_i^j}\int_{V_i^j(p)}(q-c_i^j)\frac{\partial\phi_j(q,p,t)}{\partial t}dq,$$

where $\mathcal{N}_i^j$ is the neighbor set of robot i in partition j. Similarly, $\frac{\partial m_i^j}{\partial t}$ and $\frac{\partial m_i^j}{\partial p_i}$ can be computed from (3) to obtain

$$\frac{\partial m_i^j}{\partial t} = \int_{V_i^j(p)}\frac{\partial\phi_j(q,p,t)}{\partial t}dq,$$

$$\frac{\partial m_i^j}{\partial p_i} = \int_{V_i^j(p)}\frac{\partial\phi_j(q,p,t)}{\partial p_i}dq + \int_{\partial V_i^j(p)}\phi_j(q,p,t)\sum_{l\in\mathcal{N}_i^j}\frac{(q-p_i)^\intercal}{\|p_l-p_i\|}dq.$$

Next, we provide the partial derivatives of the density function $\phi_j(q,p,t)$ required to compute the partial derivatives of m_i^j and c_i^j. For this paper, we consider static d_j^k, s_l^k, and μ_j $\forall i,j,k$. Thus, the only time-varying components of the density function are the robot positions p_l. Therefore, the partial derivatives of the density function with respect to position and time, required to compute (VII) and (VII) respectively, are

$$\frac{\partial\phi_j(q,p,t)}{\partial p_i} = -\sum_{k\in\mathcal{S},d_j^k\neq 0}\frac{2s_l^k}{\sigma}(q-p_i)^\intercal\Sigma^{-1} \cdot\exp\left[-\frac{1}{\sigma}(q-p_i)^\intercal\Sigma^{-1}(q-p_i)\right]\frac{D_j}{(S_l)^2}$$

and

$$\frac{\partial\phi_j(q,p,t)}{\partial t} = -\frac{D_j}{(S_l)^2}\sum_{l\in\mathcal{N}_i^j}\sum_{k\in\mathcal{S},d_j^k\neq 0}\frac{2s_l^k}{\sigma}(q-p_l)^\intercal\Sigma^{-1}\dot{p}_l \cdot\exp\left[-\frac{1}{\sigma}(q-p_l)^\intercal\Sigma^{-1}(q-p_l)\right].$$

References

[1] 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, 2004.

[2] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, Adaptive Coverage Control for Networked Robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[3] A. Pierson, L. C. Figueiredo, L. C. A. Pimenta, and M. Schwager, "Adapting to sensing and actuation variations in multi-robot coverage," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 337–354, 2017.

[4] A. Kwok and S. Martinez, "Energy-balancing cooperative strategies for sensor deployment," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 6136–6141.

[5] L. Pimenta, V. Kumar, R. C. Mesquita, and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control (CDC)*. IEEE, 2008, pp. 3947–3952.

[6] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Equitable partitioning policies for robotic networks," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2356–2361.

[7] J. S. Marier, C. A. Rabbath, and N. Léchevin, "Health-aware coverage control with application to a team of small uavs," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1719–1730, Sept 2013.

[8] H. Mahboubi, K. Moezzi, A. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 163–174, Feb 2014.

[9] O. Arslan and D. E. Koditschek, "Voronoi-based coverage control of heterogeneous disk-shaped robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4259–4266.

[10] A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Weighted buffered voronoi cells for distributed semi-cooperative behavior," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 5611–5617.

[11] S. Kim, M. Santos, L. Guerrero-Bonilla, A. Yezzi, and M. Egerstedt, "Coverage Control of Mobile Robots With Different Maximum Speeds for Time-Sensitive Applications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3001–3007, 2022.

[12] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage Control for Multirobot Teams With Heterogeneous Sensing Capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[13] M. Santos and M. Egerstedt, "Coverage Control for Multi-Robot Teams with Heterogeneous Sensing Capabilities Using Limited Communications," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5313–5319.

[14] A. Sadeghi and S. L. Smith, "Coverage Control for Multiple Event Types with Heterogeneous Robots," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3377–3383.

[15] M. Davoodi and J. M. Velni, "Heterogeneity-Aware Graph Partitioning for Distributed Deployment of Multiagent Systems," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2578–2588, 2022.

[16] A. McDonald, L. Wei, and V. Srivastava, "Online Estimation and Coverage Control with Heterogeneous Sensing Information," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 144–149.

[17] B. Reily, T. Mott, and H. Zhang, "Adaptation to Team Composition Changes for Heterogeneous Multi-Robot Sensor Coverage," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9051–9057.

[18] S. Choudhury, J. K. Gupta, M. J. Kochenderfer, D. Sadigh, and J. Bohg, "Dynamic multi-robot task allocation under uncertainty and temporal constraints," *Autonomous Robots*, vol. 46, no. 1, pp. 231–247, 1 2022.

[19] N. Giannakeas, K. D. Tzimourta, A. Zubiaga, A. Chatzisavvas, P. Chatzitoulousis, D. Ziouzios, and M. Dasygenis, "A Routing and Task-Allocation Algorithm for Robotic Groups in Warehouse Environments," *Information 2022, Vol. 13, Page 288*, vol. 13, no. 6, p. 288, 6 2022.

[20] G. S. Oliveira, J. Röoning, J. T. Carvalho, and P. D. M. Plentz, "Efficient Task Allocation in Smart Warehouses with Multi-Delivery Stations and Heterogeneous Robots," in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8.

[21] T. Gao, Y. Tian, and S. Bhattacharya, "Refuel Scheduling for Multi-robot Charging-on-Demand," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5825–5830, 2021.

[22] G. Notomista, S. Mayya, Y. Emam, C. Kroninger, A. Bohannon, S. Hutchinson, and M. Egerstedt, "A Resilient and Energy-Aware Task Allocation Framework for Heterogeneous Multirobot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 159–179, 2 2022.

[23] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[24] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks: variations on a theme," in *Proceedings of the 10th Mediterranean Conference on Control and Automation*, Lisbon, Portugal, 2002.

[25] S. G. 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, 2015.

[26] Y. Diaz-Mercado, S. G. Lee, and M. Egerstedt, "Human-swarm interactions via coverage of time-varying densities," in *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*. Springer International Publishing, 1 2017, pp. 357–385.

[27] J. Kennedy, A. Chapman, and P. M. Dower, "Generalized Coverage Control for Time-Varying Density Functions," in *2019 18th European Control Conference (ECC)*. IEEE, 2019.

[28] M. Santos, S. Mayya, G. Notomista, and M. Egerstedt, "Decentralized Minimum-Energy Coverage Control for Time-Varying Density Functions," in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2019, pp. 155–161.

[29] J. Cortes and M. Egerstedt, "Coordinated Control of Multi-Robot Systems: A Survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[30] G. Notomista and M. Egerstedt, "Constraint-Driven Coordinated Control of Multi-Robot Systems," in *2019 American Control Conference (ACC)*, 2019, pp. 1990–1996.

[31] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[32] P. K. Kundu, I. Cohen, and D. Dowling, "Kinematics," in *Fluid Mechanics*, 5th ed. Amsterdam: Elsevier Inc., 2012, pp. 85–88.