2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Coverage Control with Heterogeneous Robot Teams via Multi-Marginal Optimal Transport

Jerome Le Ny

Abstract— **Coverage control refers to the problem of simultaneously deploying a mobile robotic network and assigning tasks distributed in the environment to each robot. We focus on a natural extension of this problem, where tasks must be serviced by teams of robots from different classes. We leverage the connection between the assignment part of the coverage control problem and the theory of optimal transport to formulate and study a general coverage control problem by heterogeneous robotic teams, with possibly constraints on the utilization rate of each robot. The optimization of the assignment maps and of the utilization rates are shown to be convex problems, amenable to finite-dimensional deterministic or stochastic optimization methods. The optimization of the robot states or locations is subject to local minima as in the standard coverage control problem, but can be performed locally using deterministic or stochastic gradient descent, in a manner similar to Lloyd's method. Numerical simulations illustrate the flexibility of the formulation and the behavior of the algorithms.**

I. Introduction

A fundamental problem for mobile robotic networks is to devise scalable strategies that allocate the limited resources of the robots to a set of tasks to be performed. One instantiation of this problem is the coverage control problem [1], which can be interpreted as simultaneously assigning tasks in the environment to particular robots and moving the robotic network to service these tasks most efficiently. As shown in [1], a gradient descent approach based on Lloyd's method [2] leads to a distributed deployment algorithm for the robots that locally minimizes the coverage cost, with each robot assigned to the tasks in its Voronoi cell, i.e., the tasks it can service at the lowest cost among all robots.

Since [1], many extensions to the original coverage control formulation have been formulated and studied. The coverage control problem with additional load-balancing or area constraints is considered in [3], [4], which may prevent robots to be over- or under-utilized. It is shown that the task assignment subproblem is solved by generalized Voronoi diagrams, also known as power diagrams [5] when the underlying service cost is quadratic in the distance. The connection between these results and semi-discrete optimal transport (OT) [6]–[8] is made explicitly in [9], [10], which also emphasizes the computational and operational benefits of using stochastic optimization methods. More recently, a coverage control problem with equipartitioning constraint is again investigated from an OT point of view in [11]. Note that OT-based models, as well as related mean-field games and control models, have also been very actively investigated in the last decade for the control of large ensembles of agents, see, e.g., [12]–[17]. In contrast however, this paper focuses on coverage control for relatively small robotic networks, but a possibly continuous distribution of tasks.

This paper also relates to coverage control problems for robots with heterogeneous capabilities [18], [19] and to the (relatively sparse) literature on coverage control by teams of robots [20]–[22]. An application is treated in [10] with tasks of different types that can be serviced by pairs of heterogeneous robots. Jiang et al. [20] consider a coverage control problem where each task must be serviced by two robots from a homogeneous group, motivated by applications such as mobile bistatic radar systems. Their algorithm relies on higher-order Voronoi diagrams [23]. In [21], locations must be covered by different types of sensors carried by mobile robots, with a specific cost function that allows a solution directly in terms of standard Voronoi diagrams. Finally, our model also relates to the economic literature on matching for teams [24], [25], where agents from different groups form teams carrying out tasks. Although the aspect of optimizing the agent states is absent, the matching component of these models is also strongly related to OT theory.

Contributions: We formulate a new coverage control problem where each task needs to be serviced by a heterogeneous team of robots or agents. We decompose the problem into three nested optimization component: optimizing the agent-team assignment, the utilization rates of the agents, and finally the states or locations of the agents. This decomposition allows us to establish a direct link between the first component and the theory of multi-marginal OT (MMOT). In contrast to [3], [4], [10], [11], we include in the second component the possibility of optimizing the agent utilization rates, subject to convex constraints. Overall we show that these first two components are convex optimization problems that can be solved by descent methods on finite-dimensional spaces. We describe stochastic versions of these methods, which have the advantage of avoiding the computation of integrals over generalized Voronoi cells, and lead to adaptive algorithms when the task distribution is unknown and only samples are observed. The optimization over the agents' positions remains difficult, as in the original coverage control problem, but also amenable to deterministic and stochastic gradient descent methods to find locally optimal solutions.

The rest of the paper is organized as follows. A general team coverage control problem is formulated in Section II and compared to the standard coverage control problem. Section III adresses the assignment optimization component of the problem via MMOT. Section IV considers the problem

J. Le Ny is with the Dep. of Electrical Eng., Polytechnique Montreal, and with GERAD, Montreal, Canada (jerome.le-ny@polymtl.ca).
This work was supported by NSERC grant RGPIN-5287-2018.

of optimizing the agents' utilization rates and states, and Section V presents some illustrative simulation results. Finally, we conclude in Section VI. Due to space constraints, some proofs are only sketched or omitted.

Notation: For m a positive integer, $[m]$ represents the set $\{1, \ldots, m\}$. If $x_1, \ldots, x_m$ are are column vectors, then $\mathrm{col}(x_1, \ldots, x_m)$ denotes the larger column vector obtained by stacking $x_1, \ldots, x_m$. The Euclidean norm is denoted $|\cdot|$.

II. Problem Statement

A. Formulation

Consider a scenario where tasks or events are distributed according to a probability distribution μ supported on a set $\mathsf{Z} \subset \mathbb{R}^d$, for some integer d. A vector $z \in \mathsf{Z}$ is identified with a task and could include the location of the task, but also other attributes, discrete or continuous, such as its importance, type, etc. The distribution μ is assumed known for now. To service the tasks, we have n classes of agents, which can represent different types of robots, sensors, etc. Class i, for $i \in [n]$, has N_i agents, and agent j in class i has a state denoted g_i^j, for $j \in [N_i]$. This state could represent the agent's location, but also other characteristics, e.g., the quality of its sensors. Including all classes, we have $N_1 + \ldots + N_n := N$ agents. The vector $\boldsymbol{g}_i = \mathrm{col}(g_i^1, \ldots, g_i^{N_i})$ denotes the states of the agents in class i, $i \in [n]$, and $\boldsymbol{g} = \mathrm{col}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n)$ denotes the states of all agents. Let G denote the set of admissible agent states $\boldsymbol{g}$.

A task z needs to be serviced by a team of n agents, including one agent from each class. We denote $\mathsf{N} = [N_1] \times \ldots \times [N_n]$ the set of tuples, each representing a different team of n agents. Tuples in N are ordered by lexicographic order. In some classes one index could correspond to not using an agent from the class, so that one can capture the possibility of forming teams with effectively less than n members. For a team described by a tuple $\mathcal{J} := (j_1, \ldots, j_n) \in \mathsf{N}$, there is a cost $c(z, j_1, \ldots, j_n; \boldsymbol{g})$, also written more concisely $c(z, \mathcal{J}; \boldsymbol{g})$, for servicing a task z when the agents have states $\boldsymbol{g}$. In the following, we use $\mathbf{1}_{\mathcal{J}}$ to denote a vector in $\mathbb{R}^N$ with n components equal to 1 at the positions corresponding to $j_1, \ldots, j_n$, for the same ordering of components as in $\boldsymbol{g}$, and the other $N - n$ components equal to 0.

Within each class, the workload among agents should be balanced according to desired utilization rates, so that agent j in class i should be used at rate $a_i^j \geq 0$, with $\sum_{j=1}^{N_i} a_i^j = 1$ for all $i \in [n]$. This constraint can help ensure that some agents are not used excessively while others remain idle. Depending on the scenario, a_i^j may alternatively represent a probability of agent j in class i being used [9], a proportion of the workspace's volume assigned to the agent [3], etc. As for $\boldsymbol{g}$, we define $\boldsymbol{a}_i = \mathrm{col}(a_i^1, \ldots, a_i^{N_i})$ for $i \in [n]$ and $\boldsymbol{a} = \mathrm{col}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n) \in \Delta \subset \mathbb{R}^N$, where $\Delta = \Delta_1 \times \ldots \times \Delta_n$ and Δ_i is the probability simplex of dimension $N_i - 1$, for all $i \in [n]$. Let $T_i : \mathsf{Z} \to [N_i]$ denote a map assigning to each task z an agent $T_i(z)$ in class i, for $i \in [n]$. Let $\boldsymbol{T} = (T_1, \ldots, T_n) : \mathsf{Z} \to \mathsf{N}$. The constraints on utilization rates can be written $(T_i)_{\#}\mu = a_i$, for $i \in [n]$, where $(T_i)_{\#}\mu$ denotes the pushforward of the measure μ by T_i, i.e.,

$$\mu(T_i^{-1}(j)) = a_i^j, \quad \forall i \in [n], \forall j \in [N_i]. \tag{1}$$

In other words, the set $T_i^{-1}(j) \subset \mathsf{Z}$ of tasks in Z assigned to agent j in class i should have measure a_i^j. The constraints (1) on the map $\boldsymbol{T}$ are written more succinctly as $\boldsymbol{T}_{\#}\mu = \boldsymbol{a}$.

The team coverage control cost associated to agent states $\boldsymbol{g}$, utilization rates $\boldsymbol{a}$ and assignment $\boldsymbol{T}$, is then defined as

$$\mathcal{H}(\boldsymbol{g}, \boldsymbol{a}, \boldsymbol{T}) = \begin{cases} \int_{\mathsf{Z}} c(z, T_1(z), \ldots, T_n(z); \boldsymbol{g})\, d\mu(z) & \text{if } \boldsymbol{T}_{\#}\mu = \boldsymbol{a}, \\ +\infty & \text{otherwise,} \end{cases} \tag{2}$$

capturing the utilization rate constraints in the cost function. Let $A \subset \mathbb{R}^N$ be a closed convex set capturing constraints on the utilization rates. For example, we may have the freedom to choose $\boldsymbol{a}$ as long as each agent in class i is used a fraction at least $1/(2N_i)$ of the time, in which case A is defined by the inequalities $a_i^j \geq 1/(2N_i)$, for all $i \in [n]$ and $j \in [N_i]$. Then, a coverage control problem may be formulated as

$$\inf_{\boldsymbol{g} \in G} \inf_{\boldsymbol{a} \in \Delta \cap A} \inf_{\boldsymbol{T} : \mathsf{Z} \to \mathsf{N}} \mathcal{H}(\boldsymbol{g}, \boldsymbol{a}, \boldsymbol{T}), \tag{3}$$

to find the best possible task-team assignment, agent utilization rates and agent states minimizing the expected cost of servicing the tasks. One may also want to solve partial versions of Problem (3), keeping the agent states $\boldsymbol{g}$ and/or the utilization rates $\boldsymbol{a}$ fixed (i.e., G and/or A can be singletons).

B. Relation to Standard Coverage Control and Example

For $n = 1$ class, and generally under additional restrictions on the cost function and agent states, we recover the coverage control problem of [1]. Standard assumptions are that Z is compact and convex, representing locations in 2D or 3D space, and that for each agent j, $g_1^j \in \mathsf{Z}$. The cost is taken as $c(z, j; \boldsymbol{g}) = \tilde{c}(|z - g_1^j|)$, for $\tilde{c}$ a strictly increasing function, and the measure μ is assumed to have a density f. The rates $\boldsymbol{a}$ are unrestricted. The minimizations over $\boldsymbol{a}$ and $\boldsymbol{T}$ are then implicit (and do not appear in [1]), with the region $V_j = T_1^{-1}(j)$ for each j given by the Voronoi cell [23]

$$V_j = \{z \in \mathsf{Z} : |z - g_1^j| \leq |z - g_1^k|, \ \forall k\},$$

so that the overall cost function can be directly written

$$\mathcal{H}_{cov}(\boldsymbol{g}) = \sum_{j=1}^{N} \int_{V_j} \tilde{c}(|z - g_j|) f(z) dz. \tag{4}$$

In [3], [4], under similar assumptions, the coverage control problem is considered for $\boldsymbol{a}$ fixed. The map T is still implicit, because it can shown by direct arguments that the region $V_j^{\tilde{c}} = T_1^{-1}(j)$ for each j is then a generalized Voronoi cell

$$V_j^{\tilde{c}}(\boldsymbol{w}) = \{z : \tilde{c}(|z - g_1^j|) - w_j \leq \tilde{c}(|z - g_1^k|) - w_k, \ \forall k\},$$

for some vector of weights $\boldsymbol{w} \in \mathbb{R}^N$. The shape of $V_j^{\tilde{c}}(\boldsymbol{w})$ now depends on the choice of $\tilde{c}$. The fact that these results follow from OT theory is discussed in [9], [10], and generalized in Section III. More recently, [11] also discusses the case $a_1^j = 1/n$, $\forall j \in [N]$, from the OT point of view.

Next is an example illustrating the formulation.

Example 1: Suppose $\mathsf{Z} = \mathbb{R}^2 \times \{0,1\}$ (so $\mathsf{Z} \subset \mathbb{R}^3$), so that $z = (x,\tau) \in \mathsf{Z}$ represents both the location (via $x \in \mathbb{R}^2$) and the type (via $\tau \in \{0,1\}$) of the task, e.g., task of type 0 consists in measuring both the temperature and humidity at x, whereas task of type 1 requires measuring a radiation level. We have n classes of mobile sensors, and interpret the states $g_i^j \in \mathbb{R}^2$ as the sensor locations. A task of type $\tau = 0$ is associated with a service cost

$$c_0(x, j_1, \ldots, j_n; \boldsymbol{g}) = \max_{1 \le i \le n} \left\{ \left| x - \boldsymbol{g}_i^{j_i} \right|^2 \right\}, \qquad (5)$$

penalizing the largest squared distance to the task's location among the agents in the team.

On the other hand, the cost for a task with type $\tau = 1$ is

$$c_1(x, j_1, \ldots, j_n; \boldsymbol{g}) = \frac{\prod_{i=1}^n |x - g_i^{j_i}|^2}{\alpha_1 + \alpha_2 \sum_{k \neq l} |g_k^{j_k} - g_l^{j_l}|^2}, \qquad (6)$$

for some positive constants α_1, α_2, where the numerator encourages forming teams of sensors that are close to the task's location, but the denominator captures an interference cost between sensors of different classes.

Let μ_0 and μ_1 be two probability measures on $\mathbb{R}^2$ specifying the distributions of the tasks with $\tau = 0$ and $\tau = 1$ respectively, and let ν_0 and ν_1 be two positive numbers with $\nu_0 + \nu_1 = 1$, specifying the respective frequencies of tasks with $\tau = 0$ and $\tau = 1$. Then the integral cost in (2) reads

$$\sum_{i=0}^{1} \nu_i \int_{\mathbb{R}^2} c_i(x, \boldsymbol{T}(x,i); \boldsymbol{g}) d\mu_i(x)$$

for $\boldsymbol{T} : \mathsf{Z} \to \mathsf{N}$ satisfying some additional constraints (1).

III. Solving the Assignment Problem

Assume $\boldsymbol{g}$ and $\boldsymbol{a}$ in this section are fixed. The remaining minimization problem in (3) over the maps $\boldsymbol{T} : \mathsf{Z} \to \mathsf{N}$ reads

$$\begin{aligned} &\inf_{T_1,\ldots,T_n} \int_{\mathsf{Z}} c(z, T_1(z), \ldots, T_n(z); \boldsymbol{g}) \, d\mu(z). \\ &\text{s.t. } (T_i)_\# \, \mu = \boldsymbol{a}_i, \; 1 \le i \le n. \end{aligned} \qquad (7)$$

This problem is an MMOT problem [6], in its Monge formulation, i.e., the optimization variables are the measurable functions $T_i : \mathsf{Z} \to [N_i]$ for $i \in [n]$.

A. Characterization of the Monge Map

Let $\mathsf{S} := \mathsf{Z} \times \mathsf{N}$, $\Gamma(\mathsf{S})$ be the set of positive Borel measures over S, and π_{Z} denote the projection from S to Z, i.e., $\pi_{\mathsf{Z}}(z, j_1, \ldots, j_n) = z$. Similarly π_i denotes the projection from S to $[N_i]$, for $i \in [n]$. The Kantorovitch relaxation of Problem (7) above is the following optimization problem over transport plans $\gamma \in \Gamma(\mathsf{S})$

$$\begin{aligned} &\inf_{\gamma \in \Gamma(\mathsf{S})} \int_{\mathsf{S}} c(z, j_1, \ldots, j_n; \boldsymbol{g}) \, d\gamma(z, j_1, \ldots, j_n) \\ &\text{s.t. } (\pi_z)_\# \gamma = \mu, \;\; (\pi_i)_\# \gamma = \boldsymbol{a}_i, \; i \in [n]. \end{aligned} \qquad (8)$$

The constraints in (8) impose that the marginals of γ be μ, $\boldsymbol{a}_1$, $\ldots$, $\boldsymbol{a}_n$. Since these marginals are probability distributions, γ must also be a probability distribution over S. It describes the joint probability that a task is of type z and serviced by the agents $(j_1, \ldots, j_n)$. The problem (8) is a relaxation of (7), because (7) results by restricting γ to measures "induced by deterministic maps", i.e., of the form $\gamma = (\mathrm{id}_{\mathsf{Z}}, T_1, \ldots, T_n)_\# \mu$, for some maps T_i, with id_{Z} denoting the identity map on Z.

An important feature of the Kantorovitch relaxation (8) is that it is a linear programming problem, hence admits a nice duality theory [6]. Denote $L^1_\mu(\mathsf{Z})$ the set of μ-integrable functions on Z. The dual problem of (8) reads

$$\begin{aligned} &\sup_{\varphi \in L^1_\mu(\mathsf{Z}), \boldsymbol{w} \in \mathbb{R}^N} \int_{\mathsf{Z}} \varphi(z) d\mu(z) + \sum_{i=1}^n \sum_{j=1}^{N_i} w_i^j a_i^j \\ &\text{s.t } \varphi(z) + w_1^{j_1} + \ldots w_n^{j_n} \le c(z, j_1, \ldots, j_n; \boldsymbol{g}), \\ &\quad \text{for } \mu\text{-almost all } z \in \mathsf{Z}, \; \forall j_1 \in [N_1], \ldots, j_n \in [N_n]. \end{aligned} \qquad (9)$$

The sum appearing in the constraint of (9) can be rewritten more compactly as $w_1^{j_1} + \ldots w_n^{j_n} = \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}}$, for $\boldsymbol{w} \in \mathbb{R}^N$ with components ordered as for $\boldsymbol{g}$ and $\boldsymbol{a}$, and the double sum in the objective as $\boldsymbol{w}^T \boldsymbol{a}$. In view of the constraint in (9), it is sufficient to maximize over functions φ of the form

$$\varphi^c(z; \boldsymbol{w}) = \min_{\mathcal{J} \in \mathsf{N}} \{ c(z, \mathcal{J}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}} \}, \qquad (10)$$

since any feasible φ must satisfy $\varphi \le \varphi^c(\cdot; \boldsymbol{w})$ pointwise. We can then define the dual function $h^{\boldsymbol{g},\boldsymbol{a}} : \mathbb{R}^N \to \mathbb{R}$ by

$$h^{\boldsymbol{g},\boldsymbol{a}}(\boldsymbol{w}) = \int_{\mathsf{Z}} \min_{\mathcal{J} \in \mathsf{N}} \{ c(z, \mathcal{J}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}} \} \, d\mu(z) + \boldsymbol{w}^T \boldsymbol{a},$$

and rewrite the dual problem (9) more simply as the finite-dimensional optimization problem

$$\sup_{\boldsymbol{w} \in \mathbb{R}^N} h^{\boldsymbol{g},\boldsymbol{a}}(\boldsymbol{w}). \qquad (11)$$

We then have the following strong duality result.

Theorem 3.1: If $c(\cdot, \cdot; \boldsymbol{g})$ is lower semi-continuous in z and lower bounded, the infimum in (8) is attained, and its value is equal to the supremum in (11). Moreover, if Z is compact, then for any $\boldsymbol{a} \in \Delta$, the supremum in (11) is achieved for some finite vector $\boldsymbol{w}^* \in \mathbb{R}^N$.

Proof: The fact that the minimum is attained in (8) and that the optimal primal and dual values are equal follows from OT theory [6, Section 2.1]. In general, under our assumptions the maximum in the dual (9) is achieved for some extended value function $\varphi : \mathsf{Z} \to \bar{\mathbb{R}}$ and $\boldsymbol{w} \in \bar{\mathbb{R}}^N$, where $\bar{\mathbb{R}} = (\mathbb{R} \cup \{-\infty\})$ [7, Section 2.4]. Indeed, some components of $\boldsymbol{w}$ equal to $-\infty$ can occur for instance when the support of μ is unbounded and some components of $\boldsymbol{a}$ are zero. However, when Z is compact we can follow the same argument as in [17, Lemma 9] in the two-marginal case to show that an finite maximizer $\boldsymbol{w}$ exists, for any $\boldsymbol{a}$. ■

Define now for any $\boldsymbol{w} \in \mathbb{R}^N$ and any tuple $\mathcal{J} \in \mathsf{N}$ the following regions of Z

$$\begin{aligned} V^c_{\boldsymbol{g}}(\mathcal{J}; \boldsymbol{w}) := \{ &z \in \mathsf{Z} \text{ such that for all } \mathcal{L} \in \mathsf{N}, \\ &c(z, \mathcal{J}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}} \le c(z, \mathcal{L}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{L}} \}, \end{aligned} \qquad (12)$$

generalizing Voronoi cells [23]. For $\boldsymbol{w} \in \mathbb{R}^N$, for each $z \in \mathsf{Z}$, denote $\mathcal{J}_{\boldsymbol{g}}(z, \boldsymbol{w}) \in \mathsf{N}$ a tuple achieving the minimum in

$$\mathcal{J}_{\boldsymbol{g}}(z, \boldsymbol{w}) \in \arg\min_{\mathcal{L} \in \mathsf{N}} \{c(z, \mathcal{L}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{L}}\}. \tag{13}$$

When multipe tuples achieve the minimum in (13), we suppose for concreteness that we choose the smallest according to the lexicographic ordering. Then, for any given $\boldsymbol{w} \in \mathbb{R}^N$, the map $z \mapsto \mathcal{J}_{\boldsymbol{g}}(z, \boldsymbol{w})$ associates to each point in Z a unique tuple in N, i.e., defines a mapping $\mathsf{Z} \to \mathsf{N}$.

The next assumption is useful to simplify the discussion.

Assumption 1: For any pair of tuples $\mathcal{J}, \mathcal{K} \in \mathsf{N}$ and any $\boldsymbol{w} \in \mathbb{R}^N$, the set

$$\left\{ z \in \mathsf{Z} \text{ s.t. } c(z, \mathcal{J}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}} = c(z, \mathcal{K}; \boldsymbol{g}) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{L}} \right\}$$

has μ-measure zero.

The sets of Assumption 1 correspond to the intersections of the cells (12) and to elements z where the minimization (13) has multiple solutions. The assumption allows us to neglect the influence of the choice of tuple for these points on the overall cost. Next we characterize the optimal solution to (7).

Theorem 3.2: Suppose that Z is compact and that Assumption 1 holds. Let $\boldsymbol{w}^*$ be a maximizer of the dual problem (11). Then the minimum values in (7) and (8) are equal and attained. The map $z \mapsto \mathcal{J}_{\boldsymbol{g}}(z, \boldsymbol{w}^*)$ defined by (13), with ties resolved arbitrarily (e.g., by choosing the smallest tuple according to the lexicographic ordering) minimizes (7). It also induces an optimal plan for (8).

Proof: Problem (8) is a relaxation of (7), so its minimum value is no greater than that of (7) and it is sufficient to show that the plan induced by the map of the theorem achieves the minimum in (8). Pick $\boldsymbol{w}^*$ maximizing (11), which exists by Theorem 3.1 for Z compact. By strong duality [6, Proposition 2.1.5], an optimal γ must be concentrated on the set

$$\{(z, \mathcal{J}) \in \mathsf{Z} \times \mathsf{N} : \varphi^c(z, \boldsymbol{w}^*) + (\boldsymbol{w}^*)^T \mathbf{1}_{\mathcal{J}} = c(z, \mathcal{J}; \boldsymbol{g})\}.$$

By definition of φ^c in (10), this implies that for μ-almost all z, $\mathcal{J}$ must attain the minimum in (13) when there is a unique minimizer. When the minimizer is not unique, an optimal plan may randomize over the minimizing tuples, however by Assumption 1, choosing one minimizer deterministically for such z has no impact on the expected cost. This concludes the proof. ∎

B. Deterministic Optimization Approach

By Theorem 3.2, the assignment optimization problem is essentially solved by identifying a vector $\boldsymbol{w}^*$ maximizing h. This can be done using the following result.

Proposition 1: The function $h^{\boldsymbol{g},\boldsymbol{a}} : \mathbb{R}^N \mapsto \mathbb{R}$ is concave. A supergradient $\boldsymbol{s}(\boldsymbol{w}) \in \mathbb{R}^N$ of $h^{\boldsymbol{g},\boldsymbol{a}}$ at $\boldsymbol{w} \in \mathbb{R}^N$ is given by

$$\boldsymbol{s}(\boldsymbol{w}) = \boldsymbol{a} - \int_{z \in \mathsf{Z}} \mathbf{1}_{\mathcal{J}_{\boldsymbol{g}}(z,\boldsymbol{w})} \, d\mu(z), \tag{14}$$

$$\text{i.e., } s(\boldsymbol{w})_i^j = a_i^j - \mu(T_i^{-1}(j, \boldsymbol{w})), \; \forall i \in [n], \forall j \in [N_i],$$

where $T_i(\cdot, \boldsymbol{w})$ is the i^{th} map defined by (13), for $i \in [n]$.

Proof: We write h instead of $h^{\boldsymbol{g},\boldsymbol{a}}$ to simplify the notation. We have

$$h(\boldsymbol{w}) = \int_{\mathsf{Z}} \min_{\mathcal{J} \in \mathsf{N}} \{c(z, \mathcal{J}; \boldsymbol{g}) + \boldsymbol{w}^T (\boldsymbol{a} - \mathbf{1}_{\mathcal{J}})\} d\mu(z). \tag{15}$$

The function inside the integral (15) is the minimum of linear functions of $\boldsymbol{w}$, hence concave in $\boldsymbol{w}$. Because concavity is preserved by the integration, $h^{\boldsymbol{g},\boldsymbol{a}}$ is indeed concave.

For the expression of the supergradient, let $\bar{\boldsymbol{w}} \in \mathbb{R}^N$, and consider $\boldsymbol{w} \in \mathbb{R}^N$. We have by (15)

$$\begin{aligned} h(\boldsymbol{w}) &\leq \int_{\mathsf{Z}} c\left(z, \mathcal{J}_{\boldsymbol{g}}(z, \bar{\boldsymbol{w}}); \boldsymbol{g}\right) - \boldsymbol{w}^T \mathbf{1}_{\mathcal{J}_{\boldsymbol{g}}(z,\bar{\boldsymbol{w}})} \, d\mu(z) + \boldsymbol{w}^T \boldsymbol{a} \\ &= \int_{\mathsf{Z}} c(z, \mathcal{J}_{\boldsymbol{g}}(z, \bar{\boldsymbol{w}}); \boldsymbol{g}) - \bar{\boldsymbol{w}}^T \mathbf{1}_{\mathcal{J}_{\boldsymbol{g}}(z,\bar{\boldsymbol{w}})} \, d\mu(z) + \bar{\boldsymbol{w}}^T \boldsymbol{a} \\ &\quad + (\boldsymbol{w} - \bar{\boldsymbol{w}})^T \left(\boldsymbol{a} - \int_{z \in \mathsf{Z}} \mathbf{1}_{\mathcal{J}_{\boldsymbol{g}}(z,\bar{\boldsymbol{w}})} \, d\mu(z) \right), \\ &\text{i.e.,} \quad h(\boldsymbol{w}) \leq h(\bar{\boldsymbol{w}}) + (\boldsymbol{w} - \bar{\boldsymbol{w}})^T s(\bar{\boldsymbol{w}}), \; \forall \boldsymbol{w} \in \mathbb{R}^N, \end{aligned}$$

showing that $\boldsymbol{s}(\bar{\boldsymbol{w}})$ is indeed a supergradient of h at $\bar{\boldsymbol{w}}$. ∎

With the regions (12) defined and under Assumption 1, we can also rewrite the dual function as

$$\begin{aligned} h(\boldsymbol{w}) = \sum_{j_1=1}^{N_1} \dots \sum_{j_n=1}^{N_n} \int_{V_{\boldsymbol{g}}^c(j_1,\dots,j_n;w)} \Big(c(z, j_1, \dots, j_n; \boldsymbol{g}) \\ - w_1^{j_1} - \dots - w_n^{j_n} \Big) \, d\mu(z) + \sum_{i=1}^{n} \sum_{j=1}^{N_i} w_i^j a_i^j. \end{aligned}$$

Moreover, the components of the supergradient (14) of $h^{\boldsymbol{g},\boldsymbol{a}}$ can be expressed more explicitly as

$$\begin{aligned} \boldsymbol{s}(\boldsymbol{w})_i^j = a_i^j - \sum_{j_1=1}^{N_1} \dots \sum_{j_{i-1}=1}^{N_{i-1}} \sum_{j_{i+1}}^{N_{i+1}} \dots \sum_{j_n}^{N_n} \\ \mu(V_{\boldsymbol{g}}^c(j_1, \dots, j_{i-1}, j, j_{i+1}, \dots, j_n; \boldsymbol{w})). \end{aligned} \tag{16}$$

In words, for the computation using (16) of the supergradient component $i \in [n]$, $j \in [N_i]$, we "freeze" the index j for class i, and sum over the μ-probabilities of the cells obtained by letting the other indices vary. Since the intersections of these cells have probability zero by Assumption 1, this is also the probability of the union of the cells for all the teams that include agent j of class i in their composition.

Maximization of $h^{\boldsymbol{g},\boldsymbol{a}}$ to obtain an optimal vector $\boldsymbol{w}^*$ can then be performed using supergradient-based optimization methods [26, Section 7.5]. In particular, using a supergradient ascent algorithm, we initialize $\boldsymbol{w}_0 \in \mathbb{R}^N$ arbitrarily, e.g., $\boldsymbol{w}_0 = 0$, and follow the iterations,

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \gamma_k \boldsymbol{s}(\boldsymbol{w}_k), \text{ for } k \geq 0, \tag{17}$$

where γ_k are some appropriately chosen stepsizes. These iterates converge to a maximizer $\boldsymbol{w}^*$ for a variety of choices of sequence γ_k, for instance those satisfying the conditions

$$\gamma_k \geq 0, \; \sum_{k=0}^{\infty} \gamma_k = +\infty, \; \sum_{k=0}^{\infty} \gamma_k^2 < 0, \tag{18}$$

such as $\gamma_k = \frac{\alpha}{\beta + k}$ for some $\alpha, \beta > 0$. This stepsize sequence has the advantage of being very simple to implement, as it does not depend on the current value of the iterate.

C. Stochastic Optimization Approach

The computation of the supergradient (14) requires integrating μ over the generalized Voronoi cells (12), see (16). This can be difficult for general distributions and cost functions c, and moreover the number of cells increases exponentially with the number of classes. An alternative is to replace the gradient algorithm (17) by a stochastic gradient ascent algorithm. The algorithm presented next generalizes the one proposed in [9], [10] to the team assignment problem. We initialize $\boldsymbol{w}_0$ arbitrarily, e.g., taking $\boldsymbol{w}_0 = 0$. Then, we replace the expectation with respect to μ in (14) by a sample to get the stochastic iterates, for $k \geq 0$,

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \gamma_k \boldsymbol{s}_k, \text{ with } \boldsymbol{s}_k := \boldsymbol{a} - \mathbf{1}_{\mathcal{J}(z_k, \boldsymbol{w}_k)}, \qquad (19)$$

where γ_k is a sequence of stepsizes, and $\{z_k\}_{k\geq 0}$ are independent samples distributed according to μ. These elements z_k could be artificial samples generated for numerical optimization, or alternatively these samples could correspond to the sequential occurence of tasks/events in the environment according to μ, in which case (19) allows the agents to improve their assignments over time, as more events are observed [10]. In both cases the distribution μ does not need to be explicitly known, if one can generate or observe samples from it. The following result follows from standard stochastic approximation theory [10].

Proposition 2: If the stepsizes $\{\gamma_k\}_{k\geq 0}$ satisfy the condition (18) and the elements z_k are independent and distributed according to μ, then the iterates (19) converge toward a maximizer of $h^{\boldsymbol{g},\boldsymbol{a}}$ with probability one.

Further discussion of stochastic optimization for (two-marginal) OT can be found in [27]. One can use instead an averaged stochastic gradient method, or average multiple supergradient samples in (19) for each gradient step, etc.

D. Assigning Teams to Tasks

Computing the iterates (19) requires solving after each sample z_k, $k \geq 0$, the minimization problem (13) over tuples in N. Once an optimal or close to optimal $\boldsymbol{w}^*$ has been identified, the same problem has to be solved for each task z for which we want to assign a team of n agents. For a general cost function c, this requires finding the minimum of $N_1 \times \ldots \times N_n$ numbers, which scales exponentially with the number of classes. Still, in practical applications where n may be relatively small, a brute force search can be feasible. As discussed in [10], for $n = 1$, when the agents communicate via a network, a simple `floodmin` algorithm can be used to find the minimum of N numbers held by the N agents. For two or more classes however, each number involved in the minimization is associated with a specific team, so the `floodmin` algorithm needs to be implemented at the level of the teams, and each agent should be able to consider all potential teammates from other classes.

The detailed investigation of distributed algorithms to solve the minimization problem (19) for specific applications and cost structures is left for future work. For example, suppose that c can be decomposed as

$$c(z, j_1, \ldots, j_n; \boldsymbol{g}) = c_n(j_1) + \sum_{i=2}^{n} c_i(j_{i-1}, j_i), \qquad (20)$$

omitting z and $\boldsymbol{g}$ to simplify the notation. For concreteness, consider the case $n = 2$, and the fact that

$$\min_{j_1, j_2} \{c_1(j_1) + c_2(j_1, j_2)\} = \min_{j_1} \{c_1(j_1) + C_2(j_1)\},$$
$$\text{with } C_2(j_1) := \min_{j_2} \{c_2(j_1, j_2)\}.$$

For each agent j_1, $C_2(j_1)$ can be computed over the network of class 2 agents by `floodmin`. Then a `floodmin` algorithm can be executed over the network of class 1 agents for the minimization over j_1. Overall, this requires $N_1 + 1$ execution of the `floodmin` algorithm, as well as some connection between the networks of classes 1 and 2. This approach can be immediately generalized by dynamic programming to exploit the structure (20) and reduce the exponential growth of the minimization the problem.

Remark 1: If the cost function c decomposes as $c(z, j_1, \ldots, j_n; \boldsymbol{g}) = \sum_{i=1}^{n} c_i(z, j_i; \boldsymbol{g})$, then the minimization over $\boldsymbol{T}$ decomposes into n independent standard (two-marginal) OT problems, and one can leverage standard coverage control results, although the optimization over $\boldsymbol{g}$ still couples the problems.

IV. Higher Level Optimization

A. Optimization over the Agent Utilization Rates

Once the inner assignment optimization problem over $\boldsymbol{T}$ in (3) is solved, consider the reduced function

$$\mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a}) = \min_{\boldsymbol{T}:\mathsf{Z}\to\mathsf{N}} \mathcal{H}(\boldsymbol{g}, \boldsymbol{a}, \boldsymbol{T}).$$

We still fix $\boldsymbol{g}$ and now address the problem of minimizing $\mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a})$ with respect to the marginals $\boldsymbol{a}_i$, $i \in [n]$, over the convex set $\Delta \cap A$. We have the following result.

Theorem 4.1: The function $\boldsymbol{a} \mapsto \mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a})$ is convex. Moreover, suppose that Z is compact, and consider $\bar{\boldsymbol{a}}$ in Δ. Let $\bar{\boldsymbol{w}}^*$ be a maximizer of $h^{\boldsymbol{g},\bar{\boldsymbol{a}}}$ in (11) for this given $\bar{\boldsymbol{a}}$. Then $\bar{\boldsymbol{w}}^*$ is a subgradient of $\mathcal{H}_1(\boldsymbol{g}, \cdot)$ at $\bar{\boldsymbol{a}}$.

Proof: By Kantorovitch duality, $\mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a})$ is equal to (9), a supremum of linear functions of $\boldsymbol{a}$, hence convex in $\boldsymbol{a}$. Now let $\boldsymbol{a}, \bar{\boldsymbol{a}} \in \Delta$, and assume Z to be compact. Let $\boldsymbol{w}^*$ be a maximizer in (11) of $h^{\boldsymbol{g},\boldsymbol{a}}$ and $\bar{\boldsymbol{w}}^*$ be a maximizer of $h^{\boldsymbol{g},\bar{\boldsymbol{a}}}$. By strong duality and the definition of w^*

$$\mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a}) = h^{\boldsymbol{g},\boldsymbol{a}}(\boldsymbol{w}^*) \geq h^{\boldsymbol{g},\boldsymbol{a}}(\bar{\boldsymbol{w}}^*),$$

and moreover by the definition above (11) of $h^{\boldsymbol{g},\bar{\boldsymbol{a}}}$, we have $h^{\boldsymbol{g},\boldsymbol{a}}(\bar{\boldsymbol{w}}^*) = h^{\boldsymbol{g},\bar{\boldsymbol{a}}}(\bar{\boldsymbol{w}}^*) + (\bar{\boldsymbol{w}}^*)^T(\boldsymbol{a} - \bar{\boldsymbol{a}})$. Applying Kantorovitch duality again, we get

$$\mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a}) \geq \mathcal{H}_1(\boldsymbol{g}, \bar{\boldsymbol{a}}) + (\bar{\boldsymbol{w}}^*)^T(\boldsymbol{a} - \bar{\boldsymbol{a}}),$$

and since this inequality holds for any $\boldsymbol{a}$, $\bar{\boldsymbol{w}}^*$ is indeed a subgradient of $\mathcal{H}_1(\boldsymbol{g}, \cdot)$ at $\bar{\boldsymbol{a}}$. ■

Since $\Delta \cap A$ is closed and convex, we can now use a projected subgradient method to minimize $\boldsymbol{a} \mapsto \mathcal{H}_1(\boldsymbol{g}, \boldsymbol{a})$. We initialize the algorithm with some $\tilde{\boldsymbol{a}}_0 \in \Delta$. Then for any $k \geq 0$, we first project $\tilde{\boldsymbol{a}}_k$ onto $\Delta \cap A$, by solving the convex minimization problem $\min_{\boldsymbol{a} \in \Delta \cap A} \|\boldsymbol{a} - \tilde{\boldsymbol{a}}_k\|^2$, to obtain a feasible vector $\boldsymbol{a}_k$. Then we iterate as

$$\tilde{\boldsymbol{a}}_{k+1} = \boldsymbol{a}_k - \gamma_k \boldsymbol{w}_k^*,$$

where $\boldsymbol{w}_k^*$ is a maximizer of $h^{\boldsymbol{g}, \boldsymbol{a}_k}$ (computed using the methods of Section III), and γ_k is a sequence of stepsizes as discussed in Section III-B, which allows $\boldsymbol{a}_k$ to converge to a minimizer of $\mathcal{H}_1(\boldsymbol{g}, \cdot)$ as $k \to \infty$. Note that $\boldsymbol{h}^{\boldsymbol{g},\boldsymbol{a}}(\boldsymbol{w}) = \boldsymbol{h}^{\boldsymbol{g},\boldsymbol{a}}(\boldsymbol{w} + \lambda \mathbf{1}_N)$ for any constant $\lambda \in \mathbb{R}$, so we can always pick $\boldsymbol{w}_k^*$ such that $(\boldsymbol{w}_k^*)^T \mathbf{1}_N = 0$, which may simplify the projection step. Having converged to an optimum $\boldsymbol{a}^*$, a corresponding optimum assignment $\boldsymbol{T}$ is then specified by Theorem 3.2 with $\boldsymbol{w}^*$ a maximizer of $h^{\boldsymbol{g},\boldsymbol{a}^*}$.

B. Optimization over the Agent States

Finally, define the function

$$\mathcal{H}_2(\boldsymbol{g}) = \min_{\boldsymbol{a} \in \Delta \cap A} \min_{\boldsymbol{T}: \mathsf{Z} \to \mathsf{N}} \mathcal{H}(\boldsymbol{g}, \boldsymbol{a}, \boldsymbol{T}), \tag{21}$$

where for any $\boldsymbol{g}$ a minimizing $\boldsymbol{a}$ and $\boldsymbol{T}$ can be computed by following the methods outlined in the previous subsection. Then Problem (3) asks to minimize $\mathcal{H}_2$. We assume in this section that $G = \mathbb{R}^m$, for some integer m. In contrast to the previous subproblems, $\mathcal{H}_2$ is not a convex function of $\boldsymbol{g}$ in general, which is a source of computational difficulties in location optimization problems [23, Chapter 9] and in the original coverage control problem. Typically, available optimization methods only attempt to identify a local minimum.

As a heuristic, we consider the following alternating minimization approach, based on the same principle as the well-know Lloyd's method [2] used originally in [1] for coverage control. We initialize the states at some $\boldsymbol{g}_0 \in G$. Then, at given states $\boldsymbol{g}_k$ for $k \geq 0$, we compute optimal $\boldsymbol{a}_k^*$ and $\boldsymbol{T}_k^*$ minimizing (21). Fixing $\boldsymbol{a}$ and $\boldsymbol{T}$ at these values, we update $\boldsymbol{g}_k$ by following the gradient step

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \eta_k \int_{\mathsf{Z}} \frac{\partial}{\partial \boldsymbol{g}} c(z, T_{k,1}^*(z), \ldots, T_{k,n}^*(z); \boldsymbol{g}_k)\, d\mu(z), \tag{22}$$

for some stepsizes η_k, assuming that taking the derivative inside the integral is valid. If $c(z, j_1, \ldots, j_n; \boldsymbol{g})$ depends only on the states $g_i^{j_i}$, $i \in [n]$ of the agents in team $(j_1, \ldots, j_n)$, then computing $\int_{\mathsf{Z}} \frac{\partial}{\partial g_i^j} c\, d\mu$ for agent j in class i requires an integration only over the cells of the teams to which the agent belongs. Still, computing such integrals is challenging, and since they can be interpreted again as expectations with respect to μ, one can replace (22) by the stochastic gradient version

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \eta_k \frac{\partial}{\partial \boldsymbol{g}} c(z_k, T_{k,1}^*(z_k), \ldots, T_{k,n}^*(z_k); \boldsymbol{g}_k), \tag{23}$$

where $\{z_k\}_{k \geq 0}$ are independent samples distributed according to μ. We do not state a convergence result for the scheme (23) in this paper, but illustrate its behavior numerically in the next section.

Remark 2: In [3] or more recently in [11], descent methods similar to (22) are proposed for $n = 1$ and their convergence studied under some further assumptions. In [11], the rates are fixed to $a_1^j = 1/N$ (equipartition), and simulations suggest that adding of the rate constraints can help the descent method escape local minima and minimize the total coverage (4), compared to the standard Lloyd method where rate constraints are not present. Nonetheless, the original coverage control problem and the equipartitioned one have different global optima, and the gap between these optima can be significant when the standard problem benefits from cells whose μ-measure are very different. Then, adding an equipartition constraint may prevent finding good optima for the unconstrained problem, whereas the formulation (3) allows more freedom in the selection of the rates.

For example, suppose $d = 1$ and take $\tilde{c}(x) = x^2$ in (4). Let $K > 2$ and $\mu = \left(1 - \frac{1}{K}\right) \delta_0 + \frac{1}{K} \delta_K$, where δ_x denotes the Diract measure. Hence, most of the distribution is placed at 0, but a fraction $1/K$ of this distribution is placed at position K on the line. Suppose that we have two agents. The optimum coverage control cost for (4) is 0 and can be achieved by placing one agent at 0 and one agent at K. On the other hand, using properties of one-dimensional OT [8], one can show that the optimal cost for the problem with equipartition constraint and sensors placed at x_1 and $x_2 \geq x_1$ is

$$\frac{1}{2} x_1^2 + \left(\frac{1}{2} - \frac{1}{K}\right) x_2^2 + \frac{1}{K}(K - x_2)^2,$$

which is minimized by $x_1 = 0$ and $x_2 = 2$. The coverage cost (4) for this placement is $\frac{1}{K}(K-2)^2$, which tends to $+\infty$ as K to $+\infty$. So one can find examples where the optimal solution for the equipartioned problem can be arbitrarily bad for the unconstrained coverage control problem.

V. Simulations

To illustrate the overall optimization method of Section IV-B, consider a problem with $n = 2$ classes, with $N_1 = 5$ agents in class 1 and $N_2 = 3$ agents in class 2. The set Z is the square $[0, 1]^2$ in $\mathbb{R}^2$, and the agent states g_i^j correspond to positions in $\mathbb{R}^2$. The distribution μ is assumed uniform over Z. We impose the constraints $a_1^1 \geq 0.3$ and $a_2^1 \geq 0.4$ on the utilization rates of the first agent in each class, as well as $a_1^j \geq 0.1$ and $a_2^j \geq 0.2$ for all the other agents. Simulations are performed both for the maximum cost (5), and the multiplicative cost (6), in the latter case with $\alpha_1 = 1$ and $\alpha_2 = 0$. The 3 nested gradient ascent and descent algorithms are implemented using decreasing stepsizes.

The evolution of the cost function $\mathcal{H}_2$ with the number of steps of the stochastic gradient algorithm (23) is shown on Fig. 1. The cost for the initial configuration $\boldsymbol{g}$, with optimized rates, is approximately 15×10^{-2} for (5) and 9.1×10^{-3} for (6). The final costs after 100 steps are approximately 7.6×10^{-2} and 3.8×10^{-3}. Fig. 2 shows the optimal cells of the different teams, when the agents are in their initial and final locations. Recall that at each step of the iterations (23), the cells are re-optimized and in particular satisfy the rate utilization constraints. The final optimal utilization rates are $\boldsymbol{a}_1 \approx [0.3, 0.31, 0.16, 0.1, 0.13]$, $\boldsymbol{a}_2 \approx [0.4, 0.35, 0.25]$ for the cost (5), and $\boldsymbol{a}_1 \approx [0.3, 0.18, 0.15, 0.16, 0.21]^T$,

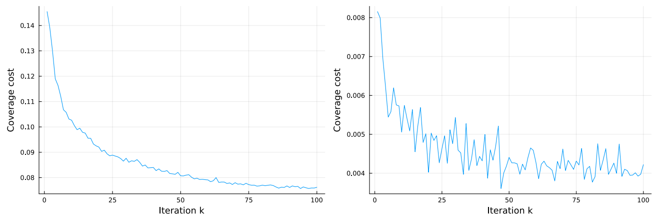

Fig. 1. Evolution of the constrained coverage cost $\mathcal{H}_2$ from (21) with the number of iterations k for the stochastic gradient descent algorithm (23). Left: assignment cost (5). Right: assignment cost (6). The same sequence of decreasing stepsizes $0.05/(1+0.5k)$ was used in both cases.

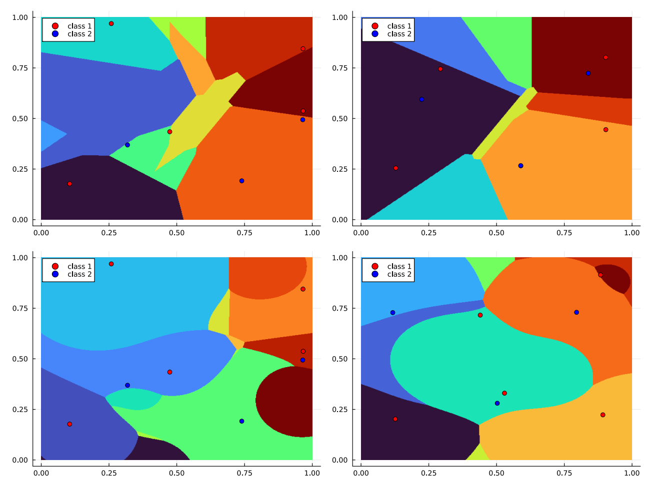

Fig. 2. Optimal rate-constrained assignment maps for the initial (left column) and final (right column) agent configurations, for the cost (5) (top row) and (6) (bottom row). The red and blue dots represent the positions of the agents in each class (on the top right figure, one blue and one red agent are superposed). Each colored region is assigned to a specific team of two agents, one from each class (so there can be at most 15 regions; the specific team assignments are not indicated in each region to preserve the readability of the figure).

$\boldsymbol{a}_2 \approx [0.4, 0.33, 0.27]^T$ for the cost (6). Despite the apparent complex shapes of the cells, the computations remain tractable when using stochastic optimization methods.

VI. Conclusion

This paper formulates and adresses a general coverage control problem for teams of agents, including the optimization of the regions assigned to the teams, of the agent utilization rates, and of the agent states. The first two optimization problems are convex and efficiently solvable, for very general underlying cost functions and task distributions, by stochastic optimization methods. The global optimization of the agent states is generally non-convex, but again local optima can be found efficiently by stochastic descent methods. A more in-depth analysis of the various convergence properties and rates for the three subproblems is left for future work.

VII. Acknowledgment

The author thanks Gero Friesecke from the Technical University of Munich for interesting discussions that inspired the work presented in this paper.

References

[1] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
[2] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[3] J. Cortés, "Coverage optimization and spatial load balancing by robotic sensor networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 749–754, 2010.
[4] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, "Distributed algorithms for environment partitioning in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1834–1848, 2011.
[5] F. Aurenhammer, "Power diagrams: properties, algorithms and applications," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 78–96, 1987.
[6] S. T. Rachev and L. Rüschendorf, *Mass Transportation Problems. Volume I: Theory*. Springer Science & Business Media, 1998, vol. 1.
[7] C. Villani, *Topics in optimal transportation*. American Mathematical Soc., 2003, vol. 58.
[8] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.
[9] J. Le Ny and G. J. Pappas, "Adaptive algorithms for coverage control and space partitioning in mobile robotic networks," *arXiv preprint arXiv:1011.0520*, 2010.
[10] ——, "Adaptive deployment of mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 654–666, 2013.
[11] D. Inoue, Y. Ito, and H. Yoshida, "Optimal transport-based coverage control for swarm robot systems: Generalization of the Voronoi tessellation-based method," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1483–1488, 2021.
[12] M. Nourian, P. E. Caines, R. P. Malhamé, and M. Huang, "Mean field LQG control in leader-follower stochastic multi-agent systems: Likelihood ratio based adaptation," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2801–2816, 2012.
[13] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *IEEE Conference on Control Applications (CCA)*, 2014, pp. 498–505.
[14] I. Haasler, J. Karlsson, and A. Ringh, "Control and estimation of ensembles via structured optimal transport," *IEEE Control Systems Magazine*, vol. 41, no. 4, pp. 50–69, 2021.
[15] V. Krishnan and S. Martínez, "A multiscale analysis of multi-agent coverage control algorithms," *Automatica*, vol. 145, 2022.
[16] Y. Chen, "Density control of interacting agent systems," *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 246–260, 2024.
[17] N. Toumi, R. Malhamé, and J. Le Ny, "Social optima of a linear quadratic collective choice model under congestion," March 2024, arXiv: 2403.10612.
[18] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE conference on decision and control*, 2008, pp. 3947–3952.
[19] Y.-F. Chung and S. S. Kia, "A distributed service-matching coverage via heterogeneous agents," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4400–4407, 2022.
[20] B. Jiang, Z. Sun, and B. D. Anderson, "Higher order Voronoi based mobile coverage control," in *American Control Conference (ACC)*, 2015, pp. 1457–1462.
[21] M. Santos and M. Egerstedt, "Coverage control for multi-robot teams with heterogeneous sensing capabilities using limited communications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5313–5319.
[22] B. Reily and H. Zhang, "Team assignment for heterogeneous multi-robot sensor coverage through graph representation learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
[23] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. Wiley & Sons, 2000.
[24] G. Carlier and I. Ekeland, "Matching for teams," *Economic theory*, vol. 42, pp. 397–418, 2010.
[25] P.-A. Chiappori, R. J. McCann, and L. P. Nesheim, "Hedonic price equilibria, stable matching, and optimal transport: equivalence, topology, and uniqueness," *Economic Theory*, vol. 42, pp. 317–354, 2010.
[26] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.
[27] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, "Stochastic optimization for large-scale optimal transport," *Advances in neural information processing systems*, vol. 29, 2016.