A Distributed Service-Matching Coverage Via Heterogeneous Agents

Yi-fan Chung and Solmaz S. Kia, *Senior Member, IEEE*

***Abstract*—We propose a distributed deployment solution for a group of networked agents that should provide a service for a large set of targets, which densely populate a finite area. The agents are heterogeneous in the sense that their quality of service (QoS), modeled as spatial Gaussian distribution, is different. To provide the best service, the objective is to deploy the agents such that their collective QoS distribution is as close as possible to the density distribution of the targets in the sense of the Kullback-Leibler divergence (KLD) measure. We propose a distributed consensus-based expectation-maximization (EM) algorithm to estimate the target density distribution, modeled as a Gaussian mixture model (GMM). Different than the existing algorithms, our proposed distributed EM algorithm enables every agent in the network to obtain an estimate of the GMM model of the distribution of the targets even if only a subset of agents can measure the targets locally. The GMM not only gives an estimate of the targets' distribution but also clusters the targets to a set of subgroups, each of which is represented by one of the GMM's Gaussian bases. We use the KLD measure to evaluate the similarity between the QoS distribution of each agent and each Gaussian basis/cluster. A distributed assignment problem is then formulated and solved as a discrete optimal mass transport problem that allocates each agent to a target cluster by taking the KLD as the assignment cost. We demonstrate our results by a sensor deployment for event detection where the sensor's QoS is modeled as an anisotropic Gaussian distribution.**

***Index Terms*—Multi-sensor deployment, distributed sensor deployment, distributed task assignment, Kullback-Leibler divergence, Gaussian mixture model.**

I. Introduction

DEPLOYING a group of sensors/agents to cover a region with service objectives such as monitoring, data collection/harvesting, and wireless communication has been of great interest in the recent decade; see for example [1]–[5]. The deployment strategy commonly includes partitioning the environment into subregions and assigning an agent to a location in each subregion such that some coverage metric is optimized. The Voronoi-based deployment strategy is a prime example of multi-agent deployment for area coverage [6]–[17]. As one of the initial work in this area, [6] developed a deployment algorithm based on the Lloyd method to compute the Voronoi partition and allocated the agents to the Centroidal Voronoi configuration, which is well-known as the optimal configuration of a class of locational optimization cost function [8]. The original Voronoi-based deployment strategy is developed for homogeneous agents. To reach the optimal coverage with heterogeneous agents whose service capabilities are different, [9]–[11] employ the weighted Voronoi diagram where the weightings account for heterogeneity among the agents. The work mentioned above assumes that the footprint of the service provided by an agent is disk-shaped, i.e., the distribution of quality of service (QoS) is isotropic. But, in practice, most sensory systems such as cameras, directional antennas, and radars are anisotropic. [12]–[14] consider, respectively, wedge-shape and elliptic service models and modify the Voronoi diagrams to match the features of the anisotropy of the sensors. But these methods increase the complexity of the Voronoi partitioning, which makes the design of distributed optimal deployment strategies very challenging. The heterogeneity in deployment algorithm design can also be due to non-uniformity in the area of interest. To deal with such scenarios, a position priority function is introduced to indicate the importance level over each location; a location needs higher QoS if the value of the priority function is higher at that location. The work [6]–[14] mentioned above assume the priority function is known to each agent. This assumption may not be realistic for every application. [15] uses the parameterized basis functions to model the priority distribution, and [16] models the distribution by a zero-mean Gaussian random field. Then, in both [15] and [16], the agents gradually fit their model to the true distribution using their local sensor measurements while exploring the area. In [17], the authors assume the unknown priority function is a function of the position of some unknown targets. The search agents aim to detect the targets while exploring the area and then broadcast their information about the environment to the service agents to decide on the deployment plan.

This paper proposes a novel distributed service-matching deployment strategy for a group of heterogeneous agents to provide a *service* to a collection of dense targets in an efficient manner by taking into account the agents' anisotropic QoS. By service, we mean serving objectives such as monitoring for event detection, data collection/harvesting, or wireless communication. The QoS of each agent is modeled as a spatial density distribution. For example, in an event detection application, the QoS can be the likelihood of event detection given the position of the targets with respect to the sensor/agent. In this paper, we model the spatial QoS of each agent as Gaussian distribution. The agent's difference in capability is captured in the size of their covariance matrix. Since the footprint of Gaussian distribution is elliptic, the agents' QoS are heterogeneous and anisotropic. In our setting, we consider the targets' density distribution, unknown a priori, as the locational priority function. The deployment objective is

Manuscript received September 10, 2021; accepted January 12, 2022. Date of publication February 7, 2022; date of current version February 24, 2022. This letter was recommended for publication by Associate Editor G. Notomista and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. This work was supported by NSF under Award IIS-1724331. *(Corresponding author: Solmaz S. Kia.)*

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92617 USA (e-mail: yfchung@uci.edu; solmaz@uci.edu).

Digital Object Identifier 10.1109/LRA.2022.3148472

to deploy the agents such that the resulting QoS distribution of the agents is similar to the density distribution of the targets. Hence, the agents' service efficiently covers the targets; i.e., the places containing more targets are served with higher QoS. We model the unknown density distribution of the targets by a Gaussian mixture model (GMM). We propose a distributed consensus-based EM algorithm to enable the agents to learn the parameters of the GMM. Different than the existing distributed EM algorithms [18], [19], our proposed distributed EM algorithm allows every agent in the network to obtain an estimate of the GMM model of the distribution of the targets even if only a subset of the agents can measure the targets locally. GMM model of the targets' distribution intrinsically clusters the targets into subgroups, each represented by a Gaussian basis of the GMM model. Therefore, after estimating the targets' density distribution, the agents also complete the targets clustering based on their spatial distribution. Our approach only requires the communication graph among the agents to be connected. This is an advantage over the Voronoi partitioning algorithms which require the Voronoi neighbor agents to communicate with each other. This requirement in Voronoi partitioning algorithms can be unrealistic because the physical distance between Voronoi neighbors may be outside the communication range of the agents. We use the KLD measure to assess the similarity of the collective QoS of the agents and the targets' density distribution. We then propose to obtain the optimal deployment pose (position and orientation) of the service agents by minimizing this KLD measure. Since this KLD measure is highly coupled and computing a distributed solution for it is challenging, we propose a suboptimal deployment solution in the form of an optimal mass transport problem. This suboptimal deployment strategy allocates each agent to a Gaussian basis/cluster of the GMM used to estimate the targets' distribution. We set the cost of transporting an agent to a target cluster as the KLD value between the agent's QoS distribution and the Gaussian basis's distribution. We show that this assignment problem can be cast as a distributed linear programming that can be solved efficiently by a distributed simplex algorithm. The GMM has been used in the robotic literature to estimate environmental factor distributions [20], [21]. The novelty in our work is to take into account the QoS of the agents and create a framework that deploys the agents using a distributed assignment approach that matches the QoS of the agents with the distribution of the targets modeled by a GMM. We illustrate our results via a sensor deployment problem for event detection.

II. Notations and Preliminaries

We let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ denote the set of real, positive real, non-negative real, integer, positive integer, and non-negative integer, respectively. For $\mathbf{s} \in \mathbb{R}^d$, $\|\mathbf{s}\| = \sqrt{\mathbf{s}^\top \mathbf{s}}$ denotes the standard Euclidean norm. We let $\mathbf{1}_n$ (resp. $\mathbf{0}_n$) denote the vector of n ones (resp. n zeros), and $\mathbf{I}_n$ denote the $n \times n$ identity matrix. Given two continuous probability density distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, the *Kullback–Leibler divergence* (KLD) is defined as $\mathrm{D}_{\mathrm{KL}}(p(\mathbf{x})||q(\mathbf{x})) = \int_{\mathbf{x} \in \mathbb{X}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x}$. KLD is a measure of similarity (dissimilarity) between two probability distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, where the smaller the value the more similar two distributions are. KLD is zero if and only if the two distribution are identical [22, p.34]. For Gaussian distributions, $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $q(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, the KLD has a closed form expression [23, eq. (2)]

$$\mathrm{D}_{\mathrm{KL}}\left(p(\mathbf{x})||q(\mathbf{x})\right) = \frac{1}{2}\Big(\ln \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} + (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) + \mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) - n\Big), \quad (1)$$

where n is the dimension of the distributions.

We follow [8] for our graph theoretic notation and definitions. A *graph* is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, \dots, N\}$ is the *node set*, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge set*, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the *adjacency* matrix such that $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $\mathsf{a}_{ij} = 0$, otherwise. An edge (i, j) from i to j means that agents i and j can communicate. A *path* is a sequence of nodes connected by edges. A *connected graph* is an undirected graph in which for every pair of nodes there is a path connecting them.

III. Problem Definition and Objective

Consider a deployment problem in which a group of $N_s \in \mathbb{Z}_{>0}$ service agents $\mathcal{V}_s$ should be deployed over a large number of targets that have densely populated a finite two-dimensional planar space $\mathcal{W}_{\mathrm{t}} \subset \mathbb{R}^2$. We let $\{\mathbf{x}_{\mathrm{t}}^n\}_{n=1}^M \subset \mathbb{R}^2$ be the set of the targets' position vector and $p(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^2$ be the density distribution of the targets, which both are *not* known to the agents a priori. The objective of the deployment can be event detection among the targets, providing wireless communication to the targets, or targets/crowd monitoring; in general we say that the agents are deployed to provide a 'service' for the targets. Let $(\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i) \in \mathbb{R}^2 \times [0, 2\pi]$ be the pose (position and orientation) of service agent $i \in \mathcal{V}_{\mathrm{s}}$. The QoS distribution $Q^i(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i)$ provided by a service agent $i \in \mathcal{V}_{\mathrm{s}}$ is modeled by a scaled Gaussian probability distribution $Q^i(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i) = z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i))$, $\mathbf{x} \in \mathbb{R}^2$, where $z^i \in \mathbb{R}_{>0}$ is the scale constant and $\mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i))$ is the Gaussian distribution with the mean $\mathbf{x}_{\mathrm{s}}^i$ and the covariance matrix $\Sigma(\theta_{\mathrm{s}}^i)$. We define the normalized collective QoS provided by the service agents by the probability density distribution

$$q(\mathbf{x}|\{\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i\}_{i \in \mathcal{V}_{\mathrm{s}}}) = \frac{\sum_{i \in \mathcal{V}_{\mathrm{s}}} Q^i}{\int_{\mathbf{x} \in \mathbb{R}^2} \sum_{i \in \mathcal{V}_{\mathrm{s}}} Q^i d\mathbf{x}} = \frac{\sum_{i \in \mathcal{V}_{\mathrm{s}}} z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i))}{\sum_{i \in \mathcal{V}_{\mathrm{s}}} z^i} = \sum_{i \in \mathcal{V}_{\mathrm{s}}} \omega_{\mathrm{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i)), \quad (2)$$

where $\omega_{\mathrm{s}}^i = \frac{z^i}{\sum_{i \in \mathcal{V}_{\mathrm{s}}} z^i}$ represents the relative service capability of agent i among $\mathcal{V}_{\mathrm{s}}$.

Th objective of this letter is to devise a deployment solution that first enables the agents to obtain an estimate $\hat{p}(\mathbf{x})$ of the density distribution of the targets in a distributed manner. Then, design a distributed deployment strategy to re-position the service agents in a way that their collective QoS serves the targets in an efficient manner. In other words, we seek locations and orientations for service agents such that the collective QoS distribution q is as much similar to as possible to the estimated target density distribution $\hat{p}$. The optimal solution for the deployment objective can be obtained from

$$\{\mathbf{x}_{\mathrm{s}}^i, \theta_{\mathrm{s}}^i\}_{i \in \mathcal{V}_{\mathrm{s}}} = \arg\min \mathrm{D}_{\mathrm{KL}}\left(\hat{p}(\mathbf{x})||q(\mathbf{x})\right). \quad (3)$$

Notice that $\hat{p}(\mathbf{x})$ and $q(\mathbf{x})$ are mixture distributions for which obtaining a closed-form for their KLD can be quite challenging. In practice, KLD for mixture models are usually estimated by using costly Monte-Carlo sampling simulations [23]. Moreover, the collective QoS distribution $q(\mathbf{x})$ contributed by each agent's

Algorithm 1: Active Weighted Average Consensus Algorithm [24], Denoted by $\{\mathbf{y}^i(\mathsf{L})\}_{i\in\mathcal{V}} \leftarrow \mathsf{ActConsen}(\{\eta^i\}_{i\in\mathcal{V}}, \{\mathsf{r}^i\}_{i\in\mathcal{V}}, \mathsf{L})$.

Require: Weight η^i, reference r^i, number of iterations L, a stepsize $\delta_c > 0$.
Initialization: $\mathbf{y}^i(0) \in \mathbb{R}^m$ and $\mathbf{v}^i(0) \in \mathbb{R}^m$
for $l = 0, 1, \ldots, \mathsf{L}$ **do**

$$\mathbf{y}^i(l+1) = \mathbf{y}^i(l) - \delta_c \eta^i \left(\mathbf{y}^i(l) - \mathsf{r}^i\right) - \delta_c \sum_{j=1}^{N} a_{ij}(\mathbf{y}^i(l) - \mathbf{y}^j(l)) - \delta_c \sum_{j=1}^{N} a_{ij}(\mathbf{v}^i(l) - \mathbf{v}^j(l)),$$

$$\mathbf{v}^i(l+1) = \mathbf{v}^i(l) + \delta_c \sum_{j=1}^{N} a_{ij}(\mathbf{y}^i(l) - \mathbf{y}^j(l)),$$

end for
return $\mathbf{y}^i(\mathsf{L})$

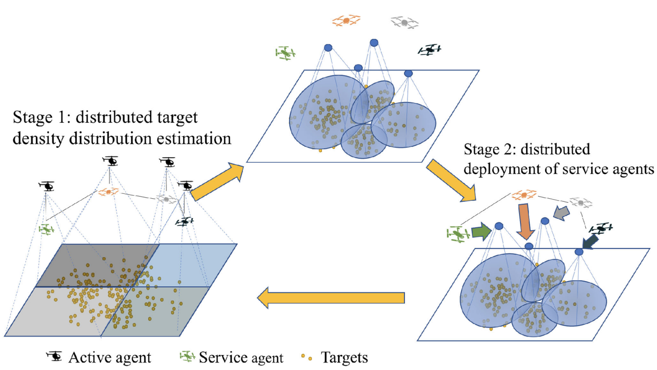

Fig. 1. The proposed two-stage distributed deployment solution.

QoS distribution, $\omega_{\mathrm{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i))$, $i \in \mathcal{V}_{\mathrm{s}}$, is a global information. Accordingly, designing a distributed solver for (3) is challenging. Therefore, in this paper, we seek a suboptimal solution for (3) that can be implemented in a distributed manner and has low computational complexity.

IV. Overview of the Proposed Agent Deployment Solution

Our proposed distributed solution to meet the objective stated in Section III is the two-stage process depicted in Fig. 1. In the first stage, we use a GMM with N_s Gaussian bases to model the targets' density distribution. Recall that N_s is the number of service agents. To provide a flexible design, we consider an operation that the agents have heterogeneous capability, some may be equipped with measurement devices to detect the targets and obtain their location, which we call them active agents, and some may be only service agent, and some may act as both active and service agent. The agents communicate over a connected undirected graph. We let $\mathcal{V}_a$ be the set of active agents. Notice that unlike some existing literature, for example [17], we do not assume that $\mathcal{V}_a$ and $\mathcal{V}_s$ are mutually exclusive. The active agents $\mathcal{V}_{\mathrm{a}}$ detect the positions of the targets, considered as the sampled data from the unknown distribution $p(\mathbf{x})$. An EM algorithm can be used to obtain the parameters of the N_s (the number of the service agents) Gaussian bases of the GMM. However, since not every agent in the network is active observers, existing distributed EM algorithms such as [18], [19] can not be used. To enable every service agent to obtain a coherent estimate of the parameters of the N_s Gaussian bases of the GMM from the measurements of active agents, we use a set of active weighted average consensus algorithms as described in Section V. For a group of agents $\mathcal{V} = \{1, \ldots, N\}$ communicating over a connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in an active weighted average consensus, only a subset of the agents $\mathcal{V}_a \subset \mathcal{V}$ are active and observe a reference value $\mathsf{r}^i \in \mathbb{R}^m$, see Fig. 2. The objective then is to enable all the agents, both active and passive, without knowing what subset of the agents are active, to obtain the weighted average of the reference values, $\frac{\sum_{j\in\mathcal{V}_a} \eta^j \mathsf{r}^j}{\sum_{j\in\mathcal{V}_a} \eta^j}$, where $\eta^j \in \mathbb{R}_{>0}$ is the weight used by active agent $j \in \mathcal{V}_a$. [24] shows that Algorithm 1, starting at any $\mathbf{y}^i(0), \mathbf{v}^i(0) \in \mathbb{R}^m$, results in $\lim_{L\to\infty} \mathbf{y}^i(\mathsf{L}) = \frac{\sum_{j\in\mathcal{V}_a} \eta^j \mathsf{r}^j}{\sum_{j\in\mathcal{V}_a} \eta^j}$ for any agent $i \in \mathcal{V}$. Notice that to simplify presentation of Algorithm 1, we substitute $\eta^i = 0$ if $i \in \mathcal{V}\backslash\mathcal{V}_a$, i.e., i is a passive agent.

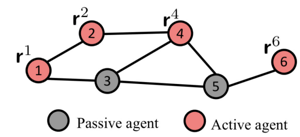

Fig. 2. An active average consensus scenario; every agent $i \in \{1, 2, 3, 4, 5, 6\}$ wants to obtain $\frac{\mathsf{r}^1+\mathsf{r}^2+\mathsf{r}^4+\mathsf{r}^6}{4}$.

The Gaussian bases of the GMM cluster the targets into N_s subgroups, each of which corresponds to a Gaussian basis. Our solution's second stage is an agent-allocation process that follows an optimal mass transport framework. In this allocation process, first each service agent $i \in \mathcal{V}_{\mathrm{s}}$ computes the KLD between its QoS distribution, $\omega_{\mathrm{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\mathrm{s}}^i, \Sigma(\theta_{\mathrm{s}}^i))$, and each cluster's Gaussian basis obtained in stage 1. A distributed assignment problem is then formulated with the KLDs as the cost of deploying the agent to each respective target cluster. As a result, each agent is paired with a target cluster, and the summation of the divergences corresponding to each paired agent's QoS distribution and target cluster's Gaussian basis is minimized. The last step in this stage is a transportation process in which a local controller can drive the agents to their assigned destinations. For dynamic targets, the process repeats. We present the details of each stage in the following sections; see Fig. 1.

V. Stage 1: Distributed Target Density Distribution Estimation

We aim to find the best mixture of Gaussian density distributions with N_s bases that describes the distribution of the targets $p(\mathbf{x})$. GMM is characterized by finite sum of Gaussian bases with different weights, means and covariance matrices. Let $\mathbf{x} \in \mathbb{R}^2$ be the observed target's position drawn from a mixture of N_s Gaussian bases with the distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$, where $\boldsymbol{\mu}_k \in \mathbb{R}^2$ is the mean and $\mathbf{\Sigma}_k \in \mathbb{R}^{2\times 2}$ is the covariance matrix for $k \in \mathcal{K} = \{1, \ldots, N_s\}$. Let $z \in \mathbb{R}$ be the indicator which indicates the variable $\mathbf{x}$ belongs to k^{th} Gaussian basis when $z = k$. The variable z is not observed so z is also called hidden variable or latent variable. The probability of drawing a variable

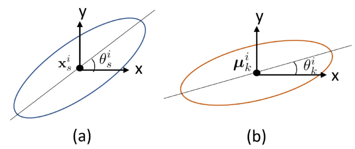

Fig. 3. The principal axis angle of (a) Agent i's QoS Gaussian distribution and (b) The k^{th} the cluster/basis of $\hat{p}^i(x)$.

from the k^{th} Gaussian basis is denoted $\pi_k := \Pr(z=k)$. The distribution of $\mathbf{x}$ given the k^{th} mixture basis is Gaussian, i.e., $\hat{p}(\mathbf{x}|z=k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Therefore, the marginal probability distribution for $\mathbf{x}$ is given by

$$\hat{p}(\mathbf{x}) = \sum_{k=1}^{N_s} \pi_k \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

The parameters that should be determined to obtain the estimate $\hat{p}(\mathbf{x})$ are the set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{N_s}$. Next, we employ the EM algorithm to obtain these parameters [25].

The EM algorithm obtains the maximum likelihood estimates of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{N_s}$ given M independent detected targets' positions $\{\mathbf{x}_{\text{t}}^n\}_{n=1}^{M}$. It is an iterative method that alternates between an expectation (E) step and a maximization (M) step. Given a detected target $\mathbf{x}_t^n$, $n \in \{1, \ldots, M\}$, E-step computes the posterior probability

$$\gamma_{kn} := \Pr(z=k|\mathbf{x}_t^n) = \frac{\pi_k \, \mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{N_s} \pi_j \, \mathcal{N}(\mathbf{x}_t^n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{5}$$

using the current value of $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{N_s}$. Then, M-step updates the parameter set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{N_s}$ by the following equations using the current γ_{kn}:

$$\pi_k = \frac{\sum_{n=1}^{M} \gamma_{kn}}{M}, \tag{6a}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{M} \gamma_{kn} \mathbf{x}_t^n}{\sum_{n=1}^{M} \gamma_{kn}}, \tag{6b}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{M} \gamma_{kn} (\mathbf{x}_t^n - \boldsymbol{\mu}_k)(\mathbf{x}_t^n - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^{M} \gamma_{kn}}, \tag{6c}$$

for $k \in \mathcal{K}$. M-step needs the global information to update the parameter set $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{N_s}$ because the summations in (6) are over all detected targets $n \in \{1, \ldots, M\}$. However, the information of the targets' positions $\{\mathbf{x}_t^n\}_{n=1}^{M}$ is distributed among the active agents $\mathcal{V}_{\text{a}}$. We observe that the right hand side quantities of (6) are in the form of (weighted) average. Hence, we propose a distributed implementation of the EM algorithm, which invokes three sets of active weighted average consensus algorithms of the form $\{\mathbf{y}^i(\mathsf{L})\}_{i\in\mathcal{V}} \leftarrow \mathsf{ActConsen}(\{\eta^i\}_{i\in\mathcal{V}}, \{\mathsf{r}^i\}_{i\in\mathcal{V}}, \mathsf{L})$, such that all the agents, $\mathcal{V} = \mathcal{V}_{\text{a}} \cup \mathcal{V}_{\text{s}}$ obtain an approximate value of (6) by locally exchanging the information with their neighbors; the approximation is due to terminating the consensus algorithm in finite time step L. The details are as follows.

Agents $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_s$ are communicating over a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$. Suppose the targets space $\mathcal{W}_{\text{t}}$ is partitioned such that each active agent is in charge of one partition in a way that their measurement zones are not overlapping, so no target is double counted in the distributed algorithm that will be employed to estimate the density distribution of the targets. Suppose each agent $i \in \mathcal{V}$ maintains a local copy of the parameter set of the Gaussian bases denoted by $\{\pi_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^{N_s}$, where the superscript shows that the variable is the local copy of agent $i \in \mathcal{V}$. At the E-step, every active agent $i \in \mathcal{V}_{\text{a}}$ computes γ_{kn} for $k \in \mathcal{K}$ and $n \in \mathcal{V}_t^i$ where $\mathcal{V}_t^i$ is the set of targets detected by active agent $i \in \mathcal{V}_{\text{a}}$. Then, in the M-step, every agent $i \in \mathcal{V}$ executes its three sets of weighted average consensus algorithms of the form $\{\mathbf{y}^i(\mathsf{L})\}_{i\in\mathcal{V}} \leftarrow \mathsf{ActConsen}(\{\eta^i\}_{i\in\mathcal{V}}, \{\mathsf{r}^i\}_{i\in\mathcal{V}}, \mathsf{L})$ as follows. In the first consensus algorithm, $\eta^i = |\mathcal{V}_t^i|$ and $\mathsf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i} \gamma_{kn}}{|\mathcal{V}_t^i|}$ if $i \in \mathcal{V}_{\text{a}}$, otherwise, $\eta^i = 0$ and $\mathsf{r}^i = 0$, results in the consensus variable $\mathbf{y}^i$ to converge close to neighborhood of (6a). In the second consensus algorithm, setting $\eta^i = \sum_{n\in\mathcal{V}_t^i} \gamma_{kn}$ and $\mathsf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i} \gamma_{kn} \mathbf{x}_n}{\sum_{n\in\mathcal{V}_t^i} \gamma_{kn}}$ if $i \in \mathcal{V}_{\text{a}}$, otherwise, $\eta^i = 0$ and $\mathsf{r}^i = 0$, results in consensus variable $\mathbf{y}^i$ converging to neighborhood of (6b). Finally, in the third consensus algorithm, setting $\eta^i = \sum_{n\in\mathcal{V}_t^i} \gamma_{kn}$ and $\mathsf{r}^i = \frac{\sum_{n\in\mathcal{V}_t^i} \gamma_{kn} (\mathbf{x}_n - \boldsymbol{\mu}_k^i)(\mathbf{x}_n - \boldsymbol{\mu}_k^i)^\top}{\sum_{n\in\mathcal{V}_t^i} \gamma_{kn}}$ if $i \in \mathcal{V}_{\text{a}}$, otherwise, $\eta^i = 0$ and $\mathsf{r}^i = 0$, results in consensus variable $\mathbf{y}^i$ to converge to neighborhood of (6c). The accuracy of the approximations of (6a), (6b) and (6c) depends on the number L of iterations of the weighted active average consensus Algorithm 1. Theoretically, if $\mathsf{L} \to \infty$ the approximation is exact because the weight η^i and the reference r^i are static in each M-step. In practice, the choice of finite L is a trade of between the accuracy of the approximation and the consumption of communication among the agents. Note that from the perspective of the weighted average consensus, how the target area $\mathcal{W}_t$ is partitioned (assuming that there is no overlap) or which agents have taken the measurements does not affect the convergence result.

By use of weighted active average consensus, all agents (passive and active) obtain an estimate on the probability distribution of the targets. But, because consensus algorithms that we use, are terminated in a finite time, it is expected that $\hat{p}^i(\mathbf{x})$ of each agent i be slightly different than other agents. In what follows we let,

$$\hat{p}^i(\mathbf{x}) = \sum_{k=1}^{N_s} \pi_k^i \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i), \tag{7}$$

be the local final estimate of agent $i \in \mathcal{V}$.

VI. Stage 2: Distributed Deployment of Service Agents

The GMM model from stage 1 intrinsically clusters the targets into a set of subgroups, each represented by a Gaussian basis of the GMM. Our suboptimal solution to the deployment problem (3) is to deploy each service agent $i \in \mathcal{V}_{\text{s}}$ to optimally cover an assigned target cluster $k \in \mathcal{K} = \{1, \ldots, N_s\}$, where $N_s = |\mathcal{V}_{\text{s}}|$. The service agent assignment is based on the similarity of the agent's QoS distribution, $\omega_{\text{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\text{s}}^i, \Sigma(\theta_{\text{s}}^i))$, to the distribution of target cluster, $\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$, such that the summation of the KLD of each assigned agent-target cluster pair is minimized. This objective can be formalized as follows. For any service agent $i \in \mathcal{V}_{\text{s}}$ let

$$C_{ik}(\mathbf{x}_{\text{s}}^i, \theta_{\text{s}}^i) = \mathrm{D}_{\mathrm{KL}}(\pi_k^i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) || \omega_{\text{s}}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_{\text{s}}^i, \Sigma(\theta_{\text{s}}^i))$$

$$= \pi_k^i \left(\ln \frac{\pi_k^i}{\omega_\mathrm{s}^i} + \mathrm{D_{KL}}\left(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)||\mathcal{N}(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \Sigma(\theta_\mathrm{s}^i))\right)\right), \tag{8}$$

for $k \in \mathcal{K}$. Notice that C_{ik} in (8) is a continuous function of the service agent's pose $(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$. We introduce a binary decision variable $Z_{ik} \in \{0,1\}$, which is 1 if agent i is assigned to region k and 0 otherwise. With the right notation at hand then, our suboptimal deployment solution is given by

$$\begin{aligned} \{\mathbf{x}_\mathrm{s}^{i\star}, \theta_\mathrm{s}^{i\star}, \{Z_{ik}^\star\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s} = \ & \arg\min \sum_{i\in\mathcal{V}_\mathrm{s}} \sum_{k\in\mathcal{K}} C_{ik}(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i) Z_{ik}, \\ & Z_{ik} \in \{0,1\}, \quad i \in \mathcal{V}_\mathrm{s}, k \in \mathcal{K}, \\ & \sum_{k\in\mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_\mathrm{s}, \\ & \sum_{i\in\mathcal{V}_\mathrm{s}} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}. \end{aligned} \tag{9}$$

Next, we introduce a set of manipulations that allows us to arrive at a distributed solution for solving (9). For each service agent $i \in \mathcal{V}_\mathrm{s}$, we start by defining

$$C_{ik}^\star = \min_{\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i} C_{ik}(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i), k \in \mathcal{K}. \tag{10}$$

Given (10) and observing that C_{ik} depends only on the pose of agent i, next we show that nonlinear mixed integer programming (9) can be cast as a linear mixed integer programming.

Lemma 1: Consider the optimization problem

$$\begin{aligned} Z_{ik}^\star = \ & \arg\min \sum_{i\in\mathcal{V}_\mathrm{s}} \sum_{k\in\mathcal{K}} C_{ik}^\star Z_{ik}, s.t. \\ & Z_{ik} \in \{0,1\}, \quad i \in \mathcal{V}_\mathrm{s}, k \in \mathcal{K}, \\ & \sum_{k\in\mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_\mathrm{s}, \\ & \sum_{i\in\mathcal{V}_\mathrm{s}} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}. \end{aligned} \tag{11}$$

where $C_{ik}^\star$ is given in (10). Let $(\mathbf{x}_\mathrm{s}^{ik\star}, \theta_\mathrm{s}^{ik\star})$ be the global minimizer of (10) for $k \in \mathcal{K}$. Let $\mathbf{x}_\mathrm{s}^{i\star}, \theta_\mathrm{s}^{i\star}$, $i \in \mathcal{V}_s$, be equal to $(\mathbf{x}_\mathrm{s}^{ik\star}, \theta_\mathrm{s}^{ik\star})$ where k corresponds to $Z_{ik}^\star = 1$. Then, $\{\mathbf{x}_\mathrm{s}^{i\star}, \theta_\mathrm{s}^{i\star}, \{Z_{ik}^\star\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s}$ is a global minimizer of optimization problem (9).

Proof: Let $f^\star = \sum_{i\in\mathcal{V}_\mathrm{s}} \sum_{k\in\mathcal{K}} C_{ik}^\star Z_{ik}^\star$. Also, let $\{\bar{\mathbf{x}}_\mathrm{s}^{i\star}, \bar{\theta}_\mathrm{s}^{i\star}, \{\bar{Z}_{ik}^\star\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s}$ be a global minimizer of (9) and $\bar{f} = \sum_{i\in\mathcal{V}_\mathrm{s}} \sum_{k\in\mathcal{K}} C_{ik}(\bar{\mathbf{x}}_\mathrm{s}^i, \bar{\theta}_\mathrm{s}^i)\bar{Z}_{ik}$. To prove our statement in what follows we show that $\bar{f} = f^\star$. Because there is no constraint on $(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$, $i \in \mathcal{V}_s$ and C_{ik} depends only on the pose of agent i, it is certain that $C_{ik}(\bar{\mathbf{x}}_\mathrm{s}^{i\star}, \bar{\theta}_\mathrm{s}^{i\star}) = C_{ik}^\star$ for k corresponding to $\bar{Z}_{ik}^\star = 1$. Consequently, since $\{\{\bar{Z}_{ik}\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s}$ satisfies constraints of (11), we can conclude that $f^\star \leq \bar{f}$. Next, notice that $\{\{Z_{ik}^\star\}_{k\in\mathcal{K}}\}_{i\in\mathcal{V}_s}$ satisfies constraints of (9). Therefore, since in (9) there is no constraint on $(\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$, $i \in \mathcal{V}_s$, it is certain that $\bar{f} \leq f^\star$, concluding the proof. □

The equivalent optimization representation (11) casts our suboptimal service agent assignment problem in the form of a discrete optimal mass transport problem [26] in which the minimum value of (8) given in (10) can be viewed as the cost of assigning agent i to the k^{th} target cluster/basis of the GMM. In Section VI-B, we show that the mixed integer programming problem (11), in fact can be cast as a linear programming in continuous space, and then solved in a distributed manner using an existing optimization algorithm. In what follows, before presenting the equivalent linear programming representation of (11), we discuss how we can obtain the minimizers of (10).

A. Similarity Between an Agent's QoS Distribution and a Target Cluster's Distribution

Given the QoS distribution provided by agent $i \in \mathcal{V}_\mathrm{s}$ to be $\omega_\mathrm{s}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i))$, where the mean of the Gaussian distribution is at the agent's location $\mathbf{x}_\mathrm{s}^i$ and the covariance matrix is with principal (major) axis at angle θ_s^i, see Fig. 3. Hence, the covariance matrix can be decomposed into $\boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i) = \mathbf{R}(\theta_\mathrm{s}^i)\boldsymbol{\Lambda}^i\mathbf{R}(\theta_\mathrm{s}^i)^\top$, where $\mathbf{R}(\theta_\mathrm{s}^i) = \begin{bmatrix} \cos\theta_\mathrm{s}^i & -\sin\theta_\mathrm{s}^i \\ \sin\theta_\mathrm{s}^i & \cos\theta_\mathrm{s}^i \end{bmatrix}$ and $\boldsymbol{\Lambda}^i = \begin{bmatrix} \sigma_x^i & 0 \\ 0 & \sigma_y^i \end{bmatrix}$, in which $\sigma_x^i, \sigma_y^i \in \mathbb{R}_{>0}$ with $\sigma_x^i \geq \sigma_y^i$ are known service parameters determines the 'shape' of the service agent i. Similarly, agent i's estimated covariance matrix $\boldsymbol{\Sigma}_k^i$, for the k^{th} target cluster/basis of its estimated $\hat{p}(x)$, see (7), can be written as $\boldsymbol{\Sigma}_k^i(\theta_k^i) = \mathbf{R}(\theta_k^i)\boldsymbol{\Lambda}_k^i\mathbf{R}(\theta_k^i)^\top$, where $\boldsymbol{\Lambda}_k^i = \begin{bmatrix} \sigma_{k,x}^i & 0 \\ 0 & \sigma_{k,y}^i \end{bmatrix}$, in which θ_k^i is the angle of principal (major) axis of the covariance matrix and $\sigma_{k,x}^i, \sigma_{k,y}^i \in \mathbb{R}_{>0}$ with $\sigma_{k,x}^i \geq \sigma_{k,y}^i$ are the variances in the major axis and minor axis direction, respectively, see Fig. 3. With the right notation at hand, the theorem below gives a closed-form solution for the minimizer $(\mathbf{x}_\mathrm{s}^{ik\star}, \theta_\mathrm{s}^{ik\star})$ of (10).

Theorem 1: Consider the optimization problem (10). Then, one of the global minimizer of optimization (10) is $(\mathbf{x}_\mathrm{s}^{ik\star}, \theta_\mathrm{s}^{ik\star}) = (\boldsymbol{\mu}_k^i, \theta_k^i)$, where θ_k^i is the angle of the principal axis of $\boldsymbol{\Sigma}_k^i$. Moreover,

$$C_{ik}^\star = \pi_k^i \left(\ln \frac{\pi_k^i}{\omega_\mathrm{s}^i} + \frac{1}{2}\left(\ln \frac{\sigma_x^i \sigma_y^i}{\sigma_{k,x}^i\, \sigma_{k,y}^i} + \frac{\sigma_{k,x}^i \sigma_y^i + \sigma_{k,y}^i \sigma_x^i}{\sigma_x^i \sigma_y^i} - 2 \right)\right). \tag{12}$$

Proof: We first note that since π_k^i and ω_s^i are fixed parameters, (10) is equivalent to minimize $\mathrm{D_{KL}}(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)||\mathcal{N}(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i)))$. From (1) we write

$$\begin{aligned} & \mathrm{D_{KL}}\left(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)||\mathcal{N}(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i))\right) \\ & = \frac{1}{2}\left(\underbrace{\ln \frac{|\boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i)|}{|\boldsymbol{\Sigma}_k^i|}}_{(a)} + \underbrace{(\mathbf{x}_\mathrm{s}^i - \boldsymbol{\mu}_k^i)^\top \boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i)^{-1}(\mathbf{x}_\mathrm{s}^i - \boldsymbol{\mu}_k^i)}_{(b)} \right. \\ & \left. \quad + \underbrace{\mathrm{tr}(\boldsymbol{\Sigma}^i(\theta_\mathrm{s}^i)^{-1}\boldsymbol{\Sigma}_k^i)}_{(c)} - 2 \right). \end{aligned} \tag{13}$$

Notice that, in (13),

$$(a) = \ln \frac{|\mathbf{R}(\theta_\mathrm{s}^i)\boldsymbol{\Lambda}^i\mathbf{R}^\top(\theta_\mathrm{s}^i)|}{|\mathbf{R}(\theta_k^i)\boldsymbol{\Lambda}_k\mathbf{R}^\top(\theta_k^i)|} = \ln \frac{|\boldsymbol{\Lambda}^i|}{|\boldsymbol{\Lambda}_k|} = \ln \frac{\sigma_x^i \sigma_y^i}{\sigma_{k,x}^i\, \sigma_{k,y}^i},$$

thus (a) does not depend on the decision variable θ_s^i. Next, notice that (b) is the only term in (13) that depends on $\boldsymbol{\mu}_k^i$ and $\mathbf{x}_\mathrm{s}^i$. For any value other than $\mathbf{x}_\mathrm{s}^i = \boldsymbol{\mu}_k^i$, (b) returns a positive value, meaning that the minimum of (13) happens at $\mathbf{x}_\mathrm{s}^{ik\star} = \boldsymbol{\mu}_k^i$. Lastly, notice that (c) in (13) reads also as

$$(c) = \mathrm{tr}(\mathbf{R}(\theta_\mathrm{s}^i)(\boldsymbol{\Lambda}^i)^{-1}\mathbf{R}(-\theta_\mathrm{s}^i + \theta_k^i)\boldsymbol{\Lambda}_k\mathbf{R}(\theta_k^i))$$

$$= \mathrm{tr}(\mathbf{R}(\theta_s^i - \theta_k^i)(\mathbf{\Lambda}^i)^{-1}\mathbf{R}(-\theta_s^i + \theta_k^i)\mathbf{\Lambda}_k).$$

Now, let $\bar{\theta} = \theta_s^i - \theta_k^i$, $s\bar{\theta} = \sin(\bar{\theta})$ and $c\bar{\theta} = \cos(\bar{\theta})$. Then, we can write (c) as

$$(c) = \mathrm{tr}\left(\begin{bmatrix} c\bar{\theta} & -s\bar{\theta} \\ s\bar{\theta} & c\bar{\theta} \end{bmatrix}\begin{bmatrix} \frac{1}{\sigma_x^i} & 0 \\ 0 & \frac{1}{\sigma_y^i} \end{bmatrix}\begin{bmatrix} c\bar{\theta} & s\bar{\theta} \\ -s\bar{\theta} & c\bar{\theta} \end{bmatrix}\begin{bmatrix} \sigma_{k,x}^i & 0 \\ 0 & \sigma_{k,y}^i \end{bmatrix}\right)$$
$$= \frac{(\sigma_{k,x}^i\sigma_y^i + \sigma_{k,y}^i\sigma_x^i)c^2\bar{\theta} + (\sigma_{k,x}^i\sigma_x^i + \sigma_{k,y}^i\sigma_y^i)s^2\bar{\theta}}{\sigma_x^i\sigma_y^i}.$$

Let $\alpha = \sigma_{k,x}^i\sigma_y^i + \sigma_{k,y}^i\sigma_x^i$ and $\beta = \sigma_{k,x}^i\sigma_x^i + \sigma_{k,y}^i\sigma_y^i$. Then, (c) reduces to

$$(c) = \frac{\alpha + (\beta - \alpha)s^2\bar{\theta}}{\sigma_x^i\sigma_y^i}.$$

Because $\sigma_{k,x}^i \geq \sigma_{k,y}^i$ and $\sigma_x^i \geq \sigma_y^i$, we have $\beta \geq \alpha$ and $(\beta - \alpha)s^2\bar{\theta}$ is non-negative. Hence, the global minimum of (c) is $\frac{\alpha}{\sigma_x^i\sigma_y^i} = \frac{\sigma_{k,x}^i\sigma_y^i + \sigma_{k,y}^i\sigma_x^i}{\sigma_x^i\sigma_y^i}$ which happens at $\bar{\theta}^\star = n\pi, n \in \{0, 1, \cdots\}$, i.e., $\theta_s^{ik^\star} = \theta_k^i + n\pi, n \in \{0, 1, \cdots\}$. To complete the proof, we note that $n = 0$ leads to one of the global minimums $\theta_s^{ik^\star} = \theta_k^i$. □

Given Theorem 1, if optimization problem (11) allocates service agent i to the k^{th} target cluster/basis of $\hat{p}^i(\mathbf{x})$, the corresponding final pose of agent i becomes $\mathbf{x}_s^{i\star} = \boldsymbol{\mu}_k^i$, $\theta_s^{i\star} = \theta_k^i$.

B. Distributed Multi-Agent Assignment Problem

The assignment optimization problem (11) is an integer optimization problem. As it is known in the discrete optimal mass transport literature [26], by the convex relaxation [27], the integer optimization (11) can be transferred to the linear programming problem stated as follows:

$$\min_{Z_{ik}\geq 0} \sum_{i\in\mathcal{V}_s}\sum_{k\in\mathcal{K}} C_{ik}^\star Z_{ik}, \text{ s.t.}$$
$$\sum_{k\in\mathcal{K}} Z_{ik} = 1, \qquad \forall i \in \mathcal{V}_s,$$
$$\sum_{i\in\mathcal{V}_s} Z_{ik} = 1, \qquad \forall k \in \mathcal{K}. \tag{14}$$

In general, problem (14) may have several optimal solutions $Z_{ik}{}^\star$s, some of them may not be integer. However, because $\mathcal{V}_s = \mathcal{K}$, as stated in [28], it is well-known that (14) has always an optimal solution $Z_{ik}^\star \in \{0, 1\}$, and that this solution corresponds exactly to the optimal assignment of (11). Since only agent i knows its own cost $C_{ik}^\star$ for $k \in \mathcal{K}$, we are interested in solving optimization problem (14) in a distributed way and such that the agents agreed on the same optimal assignment plan. The distributed simplex algorithm proposed by [28] can achieve this aim, i.e., the distributed simplex algorithm of [28] will produce a coherent optimal plan $Z_{ik}^\star \in \{0, 1\}$ across the agents. Recall that by the Brikhoff theorem [29], the extreme points (vertices) of the constraint polytop of (14) belong to $Z_{ik} \in \{0, 1\}$.

We rewrite (14) to the standard form of linear programming

$$\min_{\mathbf{Z}} \mathbf{C}^{\star T}\mathbf{Z}, \text{s.t.} \mathbf{A}\mathbf{Z} = \mathbf{b}, \quad \mathbf{Z} \geq 0. \tag{15}$$

where $\mathbf{b} = \mathbf{1}_{2N}$,

$$\mathbf{Z} = [Z_{11}, \ldots, Z_{1N}, Z_{21}, \ldots, Z_{2,N}, \ldots, Z_{N1} \cdots, Z_{NN}]^\top,$$
$$\mathbf{C}^\star = [C_{11}^\star, \ldots, C_{1N}^\star, C_{21}^\star, \ldots, C_{2,N}^\star, \ldots, C_{N1}^\star \cdots, C_{NN}^\star]^\top,$$
$$\mathbf{A} = [\mathbf{A}_{11}, \ldots, \mathbf{A}_{1N}, \mathbf{A}_{21}, \ldots, \mathbf{A}_{2,N}, \ldots, \mathbf{A}_{N1} \cdots, \mathbf{A}_{NN}],$$

in which, $\mathbf{A}_{ik} \in \mathbb{R}^{2N}$ is a column vector with i-th and $(N + k)$-th entries are 1, and others are 0. A column of problem (15) is a vector $\mathbf{h}_{ik} \in \mathbb{R}^{1+2N}$ defined as $\mathbf{h}_{ik} = [C_{ik}^\star \;\; \mathbf{A}_{ik}^\top]^\top$. The set of all columns is denote by $\mathcal{H} = \{\mathbf{h}_{ik}\}_{i\in\mathcal{V}_s, k\in\mathcal{K}}$. Thus, the linear program (15) is fully characterized by the pair $(\mathcal{H}, \mathbf{b})$. The information of $\mathcal{H}$ is distributed in the service agents. Let $\mathcal{P}^i = \{\mathbf{h}_{ik}\}_{k\in\mathcal{K}}$ is the problem column set known by agent $i \in \mathcal{V}_s$, which satisfies $\mathcal{H} = \cup_{i=1}^N \mathcal{P}^i$ and $\mathcal{P}^i \cap \mathcal{P}^j = \emptyset, \forall (i, j) \in \mathcal{V}_s$. We assume the communication graph $\mathcal{G}_s(\mathcal{V}_s, \mathcal{E}_s)$ of the service agents is connected. Hence the tuple $(\mathcal{G}_s, (\mathcal{H}, \mathbf{b}), \{\mathcal{P}^i\}_{i\in\mathcal{V}_s})$ forms a distributed linear program that can be solved by the distributed simplex algorithm [28]. The result of the optimization problem (14) is the optimal plan $Z_{ik}^\star$, where $Z_{ik}^\star = 1$ means assigning the agent i to the k^{th} target cluster with the optimal pose $\mathbf{x}_s^{i\star} = \mathbf{x}_s^{ik\star}$ and $\theta_s^{i\star} = \theta_s^{ik\star}$.

In stage 2 of our deployment solution, the last step is transporting the agents to their corresponding assigned pose. In practice, local controllers are expected to complete this task. One such local controller can be the well-known minimum energy control [30, page 138] that can transport the agents to their respective assigned pose in finite time while also enabling the agents to save on transportation energy. Lastly, note that if the targets are dynamic, our two-stage deployment process can be repeated to re-position the service agents in accordance with the changes in targets distribution.

VII. Demonstrations

We consider two sets of simulation scenarios to demonstrate the performance of the proposed distributed service-matching deployment algorithm. In these simulations, we assume that the target space $\mathcal{W}_t$ is a rectangle of $[-80, 80] \times [-60, 60]$ meters. We generate the targets from a GMM model with 12 bases so that we can evaluate the performance of our distributed EM algorithm by observing how well it estimate the original GMM distribution when we deploy 12 service agents. The number of the targets are $M = 1800$. They are shown by the colored points on Fig. 4. In the first scenario, we consider a group of 12 agents that communicate over an undirected ring graph. All the agents are service agents $\mathcal{V}_s = \{1, 2, \ldots, 12\}$ but only four agents $\mathcal{V}_a = \{2, 3, 4, 6\}$ are active agents. In the second scenario, we consider a group of 7 agents that also communicate over an undirected ring graph. All the agents are service agents $\mathcal{V}_s = \{1, 2, \ldots, 7\}$ but again only four agents $\mathcal{V}_a = \{2, 3, 4, 6\}$ are active agents. In both scenarios, we partition the target space $\mathcal{W}_t$ into four non-overlapping rectangles each assigned to one of the active agents. Active agents observe the targets in their respective space. Every agent initializes the parameters of its active average consensus algorithms locally. The initialization conditions for consensus algorithms corresponding to π_k^i is chosen randomly and for $\{\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^{N_s}$ are shown in Fig. 4. The parameters of the distributed algorithm for these scenarios are $T = 50$ and $\mathsf{L} = 50$. Agent 1's estimation results are illustrated in Fig. 4. Similar results are obtained for the other agents, but not shown here for brevity. The results show an acceptable level of performance from the distributed algorithm. The results also show the sensitivity of the EM algorithm, as it is known in the literature, to initialization. For these 2 dimensional simulations, scattering the initial guess of the agents for $\{\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\}_{k=1}^{N_s}$ uniformly, as shown in Fig. 4 for Agent 1, and also initializing

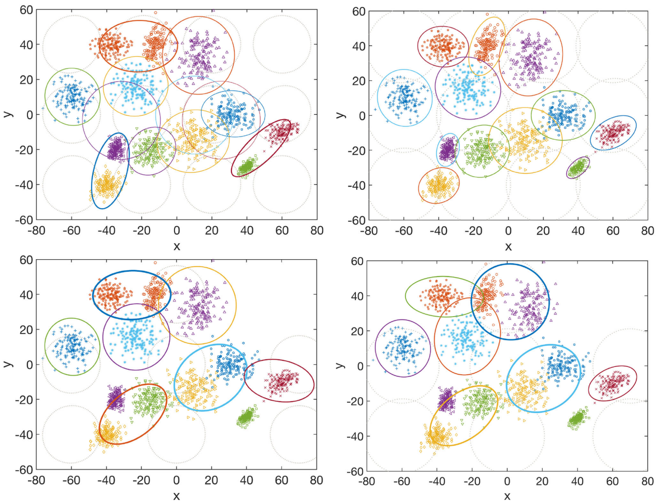

Fig. 4. The GMM estimate of Agent 1: The targets are drawn from a GMM with 12 bases; their color shows the basis they belong to. The light gray ellipses show the initial guess of Agent 1 (3σ uncertainty plot). The colored ellipses show the 3σ-plot of the bases of the estimated GMM. The top two plots show the estimate with $N_s = 12$. The bottom two plots show the results for $N_s = 7$.

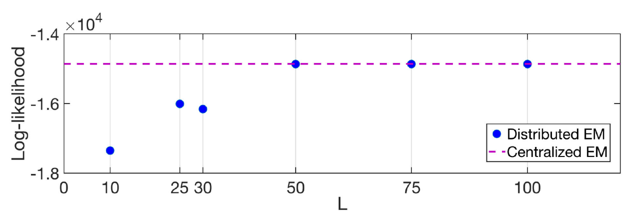

Fig. 5. The log-likelihood function for different values of L.

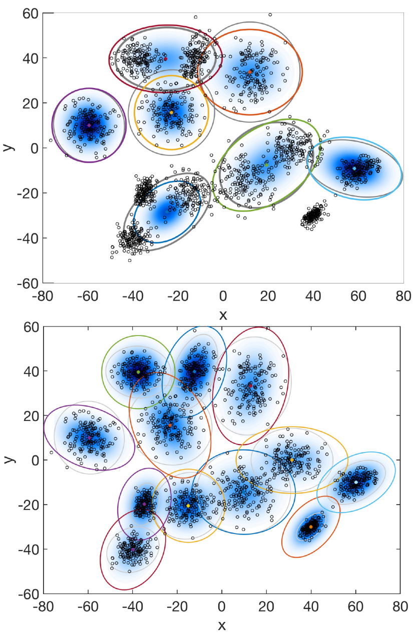

Fig. 6. The final deployment for the case with seven (top plot) and twelve (bottom plot) service agents: The gray ellipses show the 3σ GMM bases obtained by Agent 1. The colored ellipses show the 3σ plot of the QoS distribution of the service agents, while the blue color map is the collective QoS distribution (2).

with a higher level of uncertainty produces better results. The distributed EM algorithm is an approximate process of the standard centralized EM algorithm. Its accuracy depends on L, the number of iterations of consensus algorithm 1. Theoretically, if $\mathsf{L} \to \infty$ the approximation is exact because consensus algorithm converges to the exact weighted average. The choice of a finite value for L is a trade off between the accuracy of the approximation and inter-agent communication cost. Fig. 5 shows the log-likelihood, $\ln \Pr(\{\mathbf{x}_t^n\}_{n=1}^M | \{\pi_k^1 \, \boldsymbol{\mu}_k^1, \boldsymbol{\Sigma}_k^1\}_{k=1}^N)$ for agent 1 in the second simulation case of 12 service agents, in which $T = 50$ and L is varied between 10 and 100. As observed, in a modest value of $\mathsf{L} = 50$, the performance of the propose distributed EM algorithm is close to the centralized EM.

Next, suppose the service agents are equipped with a wireless sensor which is used to detect events of interest that occurred with targets. A commonly used sensor model is a probabilistic function conditioned on the sensor location and the event location [31], [32], i.e., $\Pr(\text{Detected}|\mathbf{x}_\text{s}^i, \mathbf{x}_t)$. For example in [32], given a sensor location at $\mathbf{x}_\text{s}^i, i \in \mathcal{V}_\text{s}$ and an event happening at $\mathbf{x}_t$, the probability of the sensor detecting the event is expressed as

$$\Pr(\text{Detected}|\mathbf{x}_\text{s}^i, \mathbf{x}_t) = \beta^i \mathrm{e}^{-\alpha^i \frac{(\mathbf{x}_\text{s}^i - \mathbf{x}_t)^\top (\mathbf{x}_\text{s}^i - \mathbf{x}_t)}{\gamma^{i\,2}}}$$

where $\alpha^i, \beta^i, \gamma^i$ are sensor i's parameters. In this case, the QoS of the sensor i at location $\mathbf{x}_\text{s}^i$ over the 2-D space $\mathbf{x} \in \mathbb{R}^2$ can be defined as

$$Q(\mathbf{x}|\mathbf{x}_\text{s}^i) = \Pr(\text{Detected}|\mathbf{x}_\text{s}^i, \mathbf{x}) = z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_\text{s}^i, \boldsymbol{\Lambda}^i),$$

where $z^i = \sqrt{2\pi|\boldsymbol{\Lambda}^i|}\beta^i$, $\boldsymbol{\Lambda}^i = \begin{bmatrix} \sigma^i & 0 \\ 0 & \sigma^i \end{bmatrix}$, $\sigma^i = \frac{\gamma^{i\,2}}{2\alpha^i}$. In this example, we consider a more general sensor model with anisotropic sensory capability, i.e. QoS is

$$Q(\mathbf{x}|\mathbf{x}_\text{s}^i, \theta_\text{s}^i) = z^i \mathcal{N}(\mathbf{x}|\mathbf{x}_\text{s}^i, \boldsymbol{\Sigma}^i(\theta_\text{s}^i)),$$

with $\boldsymbol{\Sigma}^i(\theta_\text{s}^i) = \mathbf{R}(\theta_\text{s}^i)\boldsymbol{\Lambda}^i\mathbf{R}^\top(\theta_\text{s}^i)$, $\boldsymbol{\Lambda}^i = \begin{bmatrix} \sigma_x^i & 0 \\ 0 & \sigma_y^i \end{bmatrix}$, and θ_s^i is the orientation of sensor i. Lastly, the collective density distribution of QoS provided by the service agents is $q(\mathbf{x}|\{\mathbf{x}_\text{s}^i, \boldsymbol{\Sigma}^i(\theta_\text{s}^i)\}_{i\in\mathcal{V}_\text{s}}) = \sum_{i\in\mathcal{V}_\text{s}} \omega_\text{s}^i \mathcal{N}(\mathbf{x}|\mathbf{x}_\text{s}^i, \boldsymbol{\Sigma}^i(\theta_\text{s}^i))$ where $\omega_\text{s}^i = \frac{z^i}{\sum_{i=1}^N z^i}$. In the first simulation scenario with 12 service agents we set $\omega_\text{s}^1 = 1/12, \sigma_x^1 = 30, \sigma_y^1 = 30$, $\omega_\text{s}^2 = 1/12, \sigma_x^2 = 30, \sigma_y^2 = 15$, $\omega_\text{s}^3 = 1/12, \sigma_x^3 = 80, \sigma_y^1 = 30$, $\omega_\text{s}^4 = 1/12, \sigma_x^4 = 70, \sigma_y^4 = 25$, $\omega_\text{s}^5 = 1/12, \sigma_x^5 = 30, \sigma_y^5 = 30$, $\omega_\text{s}^6 = 1/12, \sigma_x^6 = 60, \sigma_y^6 = 40$, $\omega_\text{s}^7 = 1/12, \sigma_x^7 = 50, \sigma_y^7 = 20$, $\omega_\text{s}^8 = 1/12, \sigma_x^8 = 30, \sigma_y^8 = 70, \omega_\text{s}^9 = 1/12, \sigma_x^9 = 40, \sigma_y^9 = 15, \omega_\text{s}^{10} = 1/12, \sigma_x^{10} = 10, \sigma_y^{10} = 30$, $\omega_\text{s}^{11} = 1/12, \sigma_x^{11} = 20, \sigma_y^{11} = 40$, $\omega_\text{s}^{12} = 1/12, \sigma_x^{12} = 20, \sigma_y^{12} = 50$. In the second scenario, we use the values for the first 7 agents listed in first scenario, with $\omega_\text{s}^i = 1/7$, $i \in \{1, \ldots, 7\}$. Each agent $i \in \mathcal{V}_\text{s}$ evaluates its costs $C_{ik}^\star$ for all $k \in \mathcal{K}$ by (12). Then, the agents cooperatively solve the distributed multi-agent assignment problem (15) by the means of distributed simplex algorithm [28]. The optimal assignment plan of (15) for the first scenario is $Z_{1,4}^\star = 1$, $Z_{2,12}^\star = 1$, $Z_{3,8}^\star = 1$, $Z_{4,11}^\star = 1$, $Z_{5,6}^\star = 1$, $Z_{6,2}^\star = 1$, $Z_{7,9}^\star = 1$, $Z_{8,10}^\star = 1$, $Z_{9,7}^\star = 1$, $Z_{10,3}^\star = 1$, $Z_{11,1}^\star = 1$, and $Z_{12,5}^\star = 1$. The assignment solution for the second scenario is $Z_{1,3}^\star = 1$, $Z_{2,1}^\star = 1$, $Z_{3,5}^\star = 1$, $Z_{4,7}^\star = 1$, $Z_{5,4}^\star = 1$, $Z_{6,2}^\star = 1$ and $Z_{6,6}^\star = 1$. The density distribution of QoS provided by the service agents (sensors) after deployment is illustrated in Fig. 6, where the black circles are the targets, the colored dots represent the service agents. The colored ellipses show the 3σ plot of the QoS distribution of the service agents, while the blue color map is the collective QoS distribution (2). The darker blue indicates a better QoS. These simulation results show that the collective QoS distribution is similar to the targets' distribution, indicating that the distribution of QoS efficiently covers the targets.

VIII. Conclusion

This paper considered the problem of distributed deployment of a group of agents to provide a service for a dense set of targets whose spatial distribution in the space was not known in advance. The quality of the service of each agent was modeled as a spatial Gaussian distribution. To solve this problem, we proposed a two-stage deployment strategy. First, we proposed a distributed consensus-based EM algorithm to enable the agents, regardless of whether they observe any target or not, to obtain an estimate of the spatial distribution of the targets as a GMM. Then, we defined the deployment objective as deploying the agents such that their collective QoS distribution is as similar as possible to the targets' density distribution. We used the KLD to measure the similarity of the distributions. Since similarity equation between the collective QoS distribution and the targets' distribution was highly coupled and computing a distributed solution for minimizing it was challenging, we proposed a suboptimal deployment solution in the form of an optimal mass transport problem to allocate each agent to a Gaussian basis of the GMM that estimated the targets' distribution. The idea was originated from observing that the GMM bases in fact cluster the targets into a set of subgroups. We can then deploy the agents in a way that each agent serves one of these subgroups. We defined the cost of transporting an agent to a target cluster as the KLD value between the agent's QoS distribution and that cluster's distribution. We showed that this assignment problem can be cast as a distributed linear programming, which can be solved efficiently by a distributed simplex algorithm to give us the final deployment locations. We illustrated our results via a set of simulations for a sensor deployment problem for event detection.

References

[1] J. Guo, P. Walk, and H. Jafarkhani, "Optimal deployments of UAVs with directional antennas for a power-efficient coverage," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 5159–5174, Aug. 2020.

[2] Q. Zhang, W. Saad, M. Bennis, X. Lu, M. Debbah, and W. Zuo, "Predictive deployment of UAV base stations in wireless networks: Machine learning meets contract theory," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 637–652, Jan. 2021.

[3] M. Kumar, K. Cohen, and B. HomChaudhuri, "Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires," *J. Aerosp. Comput., Inf., Commun.*, vol. 8, no. 1, pp. 1–16, 2011.

[4] A. Faustine, A. N. Mvuma, H. J. Mongi, M. C. Gabriel, A. J. Tenge, and S. B. Kucel, *Wireless Sensor Networks for Water Quality Monitoring and Control Within Lake Victoria Basin*. Scientific Research Publishing Inc., 2014.

[5] S. Susca, F. Bullo, and S. Martinez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, pp. 288–296, Mar. 2008.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Automat.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[7] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.

[8] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*, vol. 27. William St. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[9] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 3947–3952.

[10] A. Pierson, L. C. Figueiredo, L. C. Pimenta, and M. Schwager, "Adapting to performance variations in multi-robot coverage," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 415–420.

[11] O. Arslan and D. E. Koditschek, "Voronoi-based coverage control of heterogeneous disk-shaped robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4259–4266.

[12] K. Laventall and J. Cortés, "Coverage control by multi-robot networks with limited-range anisotropic sensory," *Int. J. Control*, vol. 82, no. 6, pp. 1113–1121, 2009.

[13] F. Farzadpour, X. Zhang, X. Chen, and T. Zhang, "On performance measurement for a heterogeneous planar field sensor network," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics*, 2017, pp. 166–171.

[14] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, "Voronoi based coverage control with anisotropic sensors," in *Proc. Amer. Control Conf.*, 2008, pp. 736–741.

[15] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robot. Res.*, vol. 28, no. 3, pp. 357–375, 2009.

[16] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, "Multi-agents adaptive estimation and coverage control using gaussian regression," in *Proc. Eur. Control Conf.*, 2015, pp. 2490–2495.

[17] F. Sharifi, M. Mirzaei, Y. Zhang, and B. W. Gordon, "Cooperative multi-vehicle search and coverage problem in an uncertain environment," *Unmanned Syst.*, vol. 3, no. 01, pp. 35–47, 2015.

[18] R. D. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

[19] P. Forero, A. Cano, and G. Giannakis, "Consensus based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 1989–1992.

[20] E. Pignat and S. Calinon, "Bayesian Gaussian mixture model for robotic policy imitation," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4452–4458, Oct. 2019.

[21] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1715–1721, Apr. 2019.

[22] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. St. New York, NY, USA: Cambridge Univ. Press, 2003.

[23] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2007, pp. IV–317.

[24] Y.-F. Chung and S. S. Kia, "Dynamic active average consensus," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1177–1182, Oct. 2021.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc.: Ser. B*, vol. 39, no. 1, pp. 1–22, 1977.

[26] G. Peyré and M. Cuturi, "Computational optimal transport," *Found. Trends Mach. Learn.*, vol. 11, no. 5-6, pp. 355–607, 2019.

[27] D. P. Bertsekas, *Network optimization: Continuous and discrete models*. Belmont, MA, USA: Athena Scientific, 1998.

[28] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, "A distributed simplex algorithm for degenerate linear programs and multi-agent assignments," *Automatica*, vol. 48, no. 9, pp. 2298–2304, 2012.

[29] G. Birkhoff, "Tres observaciones sobre el álgebra lineal," *Univ. Nac. Tucumán. Revista A*, vol. 5, pp. 147–151, 1946.

[30] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control.*, River Street, Hoboken, NJ: Wiley, 2012.

[31] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," in *Field and Service Robotics*. Lake Yamanaka, Japan: Springer, 2003, pp. 209–222.

[32] H. Xiao, R. Cui, and D. Xu, "A sampling-based bayesian approach for cooperative multiagent online search with resource constraints," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1773–1785, Jun. 2018.