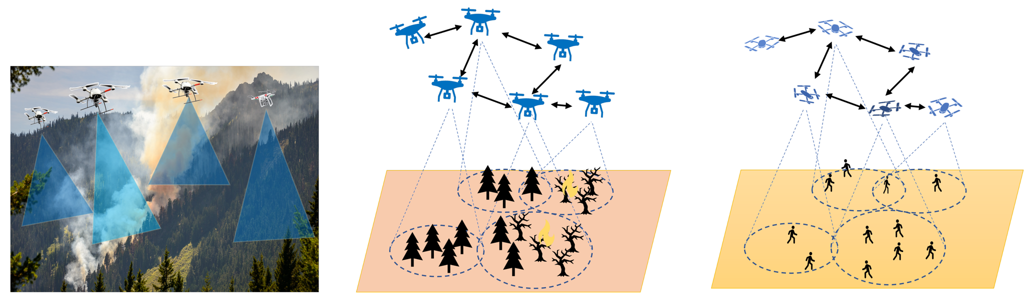
A Multi-agent Service-Matching Deployment for Agents with Probabilistic Service Coverage

Donipolo Ghimire and Solmaz S. Kia

Cooperative Systems Lab, Mechanical and Aerospace Eng. Dept. University of California Irvine

Objective statement

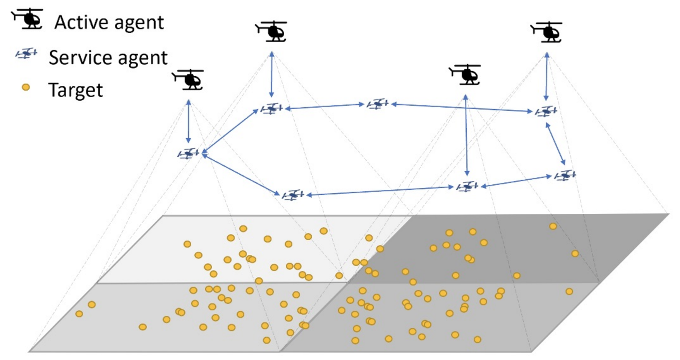
We seek a solution for a group of networked agents that should provide a service for a large set of targets, which densely populate a finite area.



- The distribution of targets is not known a priori
- The agent's quality of service (QoS) is modeled as spatial Gaussian distribution and service here refers to events like event detection, monitoring, wireless coverage etc.

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. We use Wasserstein distance('earths mover distance') as the measure of comparison between the QoS and target distribution.

Problem settings

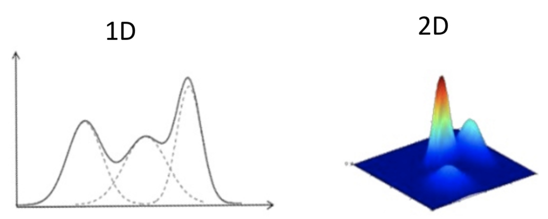


- $p(\mathbf{x})$: the density distribution of targets is not known to the agents
- Active agents only observe the targets
- Service agents provide the service
- Active and service agents communicate over a connected graph
- Active and service agent sets can overlap
- $\mathbf{x}_\mathrm{s}^i$ is the location of the projection of UAV i onto the ground
- QoS provided by a service agent ($i \in V_\mathrm{s}$) is
  - $Q(\mathbf{x}|\mathbf{x}_\mathrm{s}^i) = \omega_\mathrm{s}^i q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$, where $q^i(\mathbf{x}|\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i)$ can be a non-Gaussian or density Gaussian depending on the objective in hand, where $w^i \in$ is the scale constant.

Solution approach

- Model the unknown target density distribution by Gaussian Mixture Model (GMM)

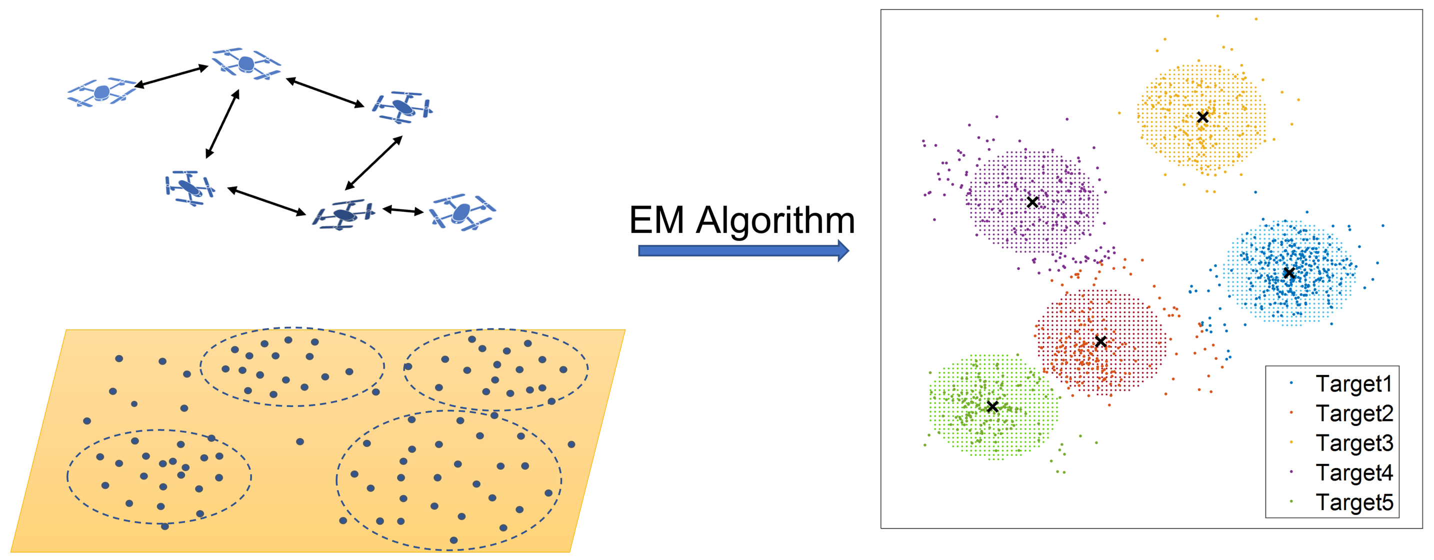
$$\hat{p}(\mathbf{x}) = \sum_{k=1}^{N} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



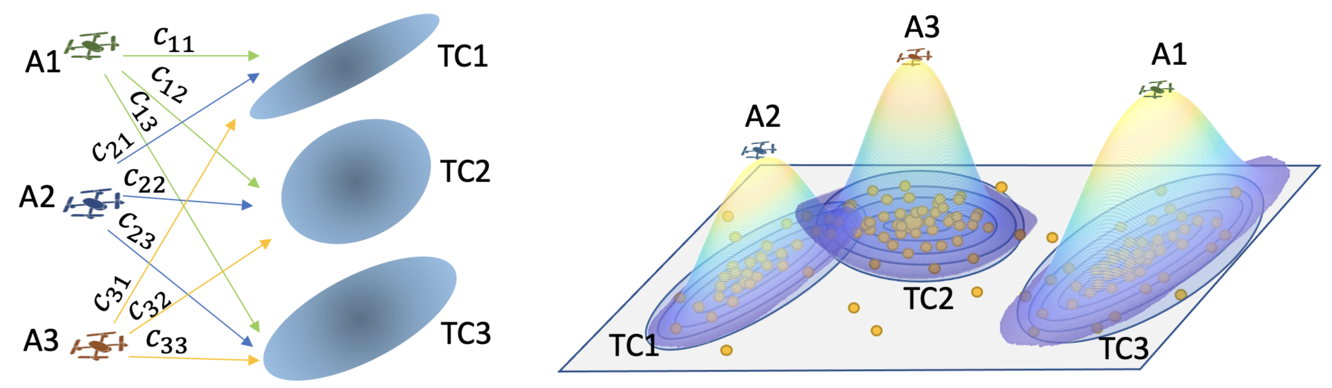
- Perform convolution inspired operation between discrete target distribution $p(\mathbf{x})$ and QoS $Q(\mathbf{x}|\mathbf{x}_\mathrm{s}^i)$, where the operation is calculating the Wasserstein metric between two the discrete $p(\mathbf{x})$ and the discrete $Q(\mathbf{x}|\mathbf{x}_\mathrm{s}^i)$ (both represent as a matrix) to create a Cost map $\mathcal{R}$
- Use a Sequential greedy algorithm to pick the best spots in the reward map for i Agents.

The proposed framework

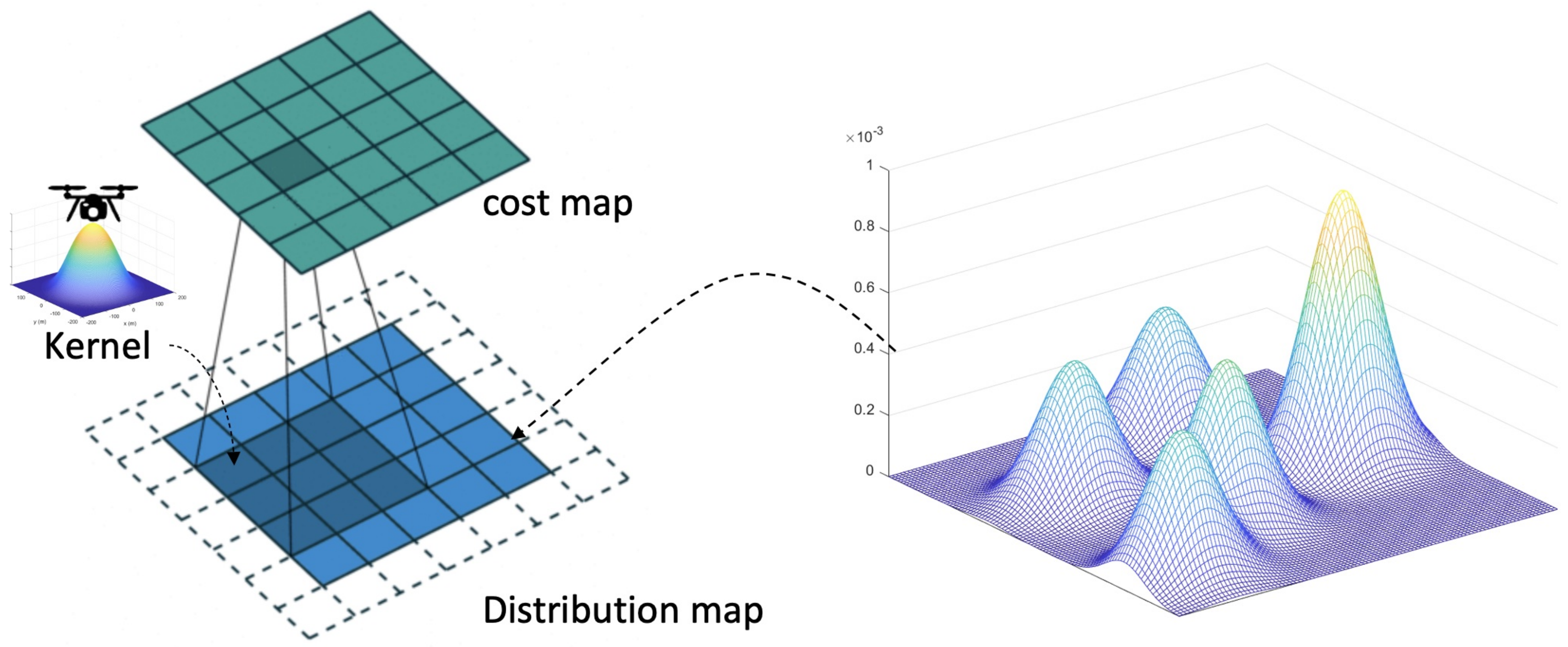
- Model the Target density using Gaussian Mixture Model



- One possible solution: a service matching deployment as optimal linear task assignment: $c_{ij} = KLD(Q_i, B_i)$



- Our proposed solution: Convolution inspired reward map generation followed by sequential greedy selection



Convolution rule

Fixed-orientation deployment

- The Wasserstein metric between two normalized discrete histograms with N bins is simplified to the ℓ_1 distance.

$$\mathcal{W}_{1d_{i,j}\neq 0}(\mathbf{q}, \bar{\mathbf{p}}) = \frac{1}{2}\sum_{i=0}^{N-1} |\mathrm{q}_i - \bar{\mathrm{p}}_i| = \frac{1}{2}||\mathbf{q} - \bar{\mathbf{p}}||_1 \quad (1)$$

where $||\cdot||_1$ is the discrete ℓ_1 norm.

- Once the above operation is performed between target distribution $p(\mathbf{x})$ and QoS $Q(\mathbf{x}|\mathbf{x}_\mathrm{s}^i)$

$$\mathcal{R}((\mathbf{x}_\mathrm{s}^i))^\star = \min_{\mathbf{x}_\mathrm{s}^i, \theta_\mathrm{s}^i} \mathcal{W}_{1d_{i,j}\neq 0}(\mathbf{q}, \bar{\mathbf{p}}) \quad (2)$$

Controllable-orientation deployment

- Rotate the kernel according to a set of pre-specified rotation angles
- At each angle compute the Wasserstein metric
- Choose the orientation with the smallest Wasserstein metric value as the cost for that given cell
- The deployment position and orientation of a robot is determined by the position and orientation of the cell with the smallest cost value.

Methodology contd.

- Sequential Greedy Algorithm

Algorithm 1 Sequential Greedy Algorithm

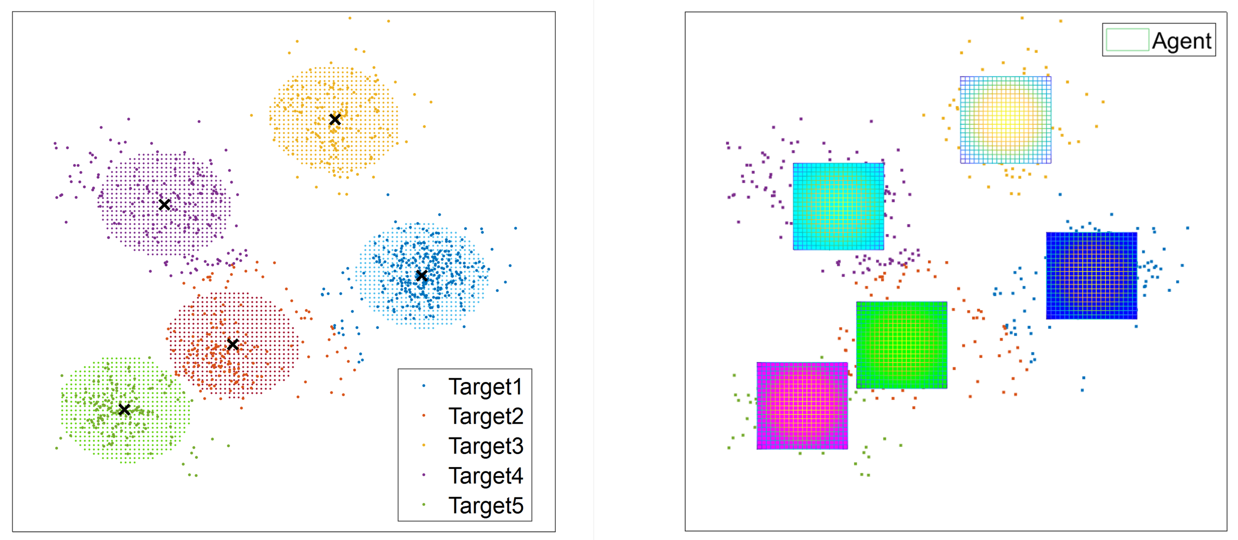
1: **procedure** SGOpt(Π^i, $i \in \mathcal{A}$)
2: Init: $\bar{\Pi}^i \leftarrow \emptyset$, $i \leftarrow 0$
3: **for** $i \in \mathcal{A}$ **do**
4: $\Pi^{i\star} = \underset{\pi \subset \Pi^i}{\mathrm{argmin}}\ \mathcal{R}(\mathbf{x}_\mathrm{s}^i)$
5: $\bar{\Pi} \leftarrow \bar{\Pi} \cup \pi^{i\star}$.
6: **end for**
7: **Return** $\bar{\Pi}$.
8: **end procedure**

Numerical examples

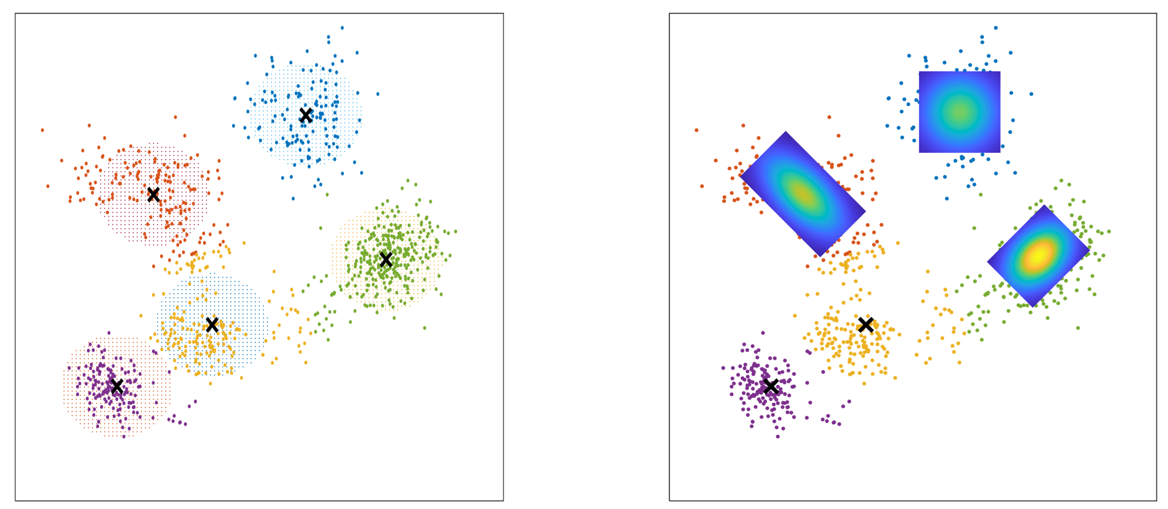
- UAV-aid wireless communication to users on the ground: The QoS of the UAV i to user at $\mathbf{x}_t$

$$\Pr(\Gamma \geq \zeta) = \mathrm{Q}(\sqrt{2K}, \sqrt{2\zeta(1+K)d^\alpha/\beta^i}), \quad (3)$$

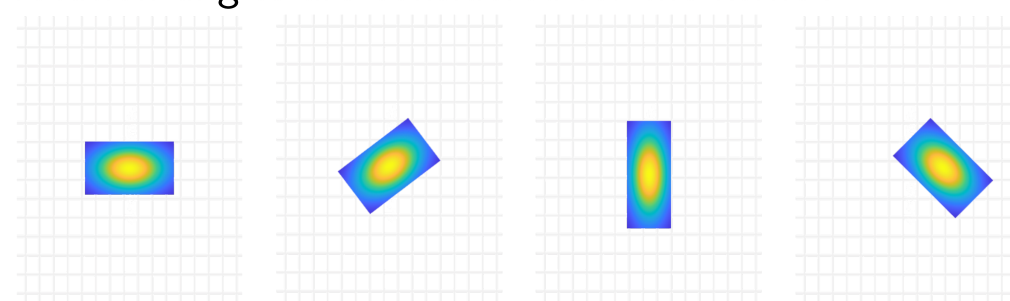
where Q is the first-order Marcum $Q-$ function and $\beta^i = \frac{\gamma^i P_T^i}{N_0^i}$



- Event detection where detection capability modeled as directional Gaussian distribution $\Pr(\text{Detected}|\mathbf{x}_\mathrm{s}^i, \mathbf{x}_t) = \beta^i \mathrm{e}^{-\alpha^i \frac{(\mathbf{x}_\mathrm{s}^i - \mathbf{x}_t)^\top (\mathbf{x}_\mathrm{s}^i - \mathbf{x}_t)}{\gamma^{i2}}}$



Four rotation angles are considered for the kernel.



Corresponding author: dghimire@uci.edu