# Multi-Agent Exploration-Based Search for an Unknown Number of Targets [⋆]

**Bilal Yousuf, Zsófia Lendek, Lucian Buşoniu**

*Technical University of Cluj-Napoca, Romania*
*e-mail: {bilal.yousuf, zsofia.lendek, lucian.busoniu}@aut.utcluj.ro*

**Abstract:** This paper presents an active sensor fusion technique for multiple mobile agents (robots) to detect an unknown number of static targets at unknown positions. To process and fuse sensor measurements from the agents, we use a random finite set formulation with an iterated-corrector probability hypothesis density filter. Our main contribution is to introduce two different multi-agent planners to quickly find the targets. The planners make greedy decisions for the next state of each agent by maximizing an objective function consisting of target refinement and exploration components. We demonstrate the performance of our approach through a series of simulations using homogeneous and heterogeneous agents. The results show that our framework works better than a lawnmower baseline, and that a centralized version of the planner works best.

*Keywords:* Multi-agent system, sensor fusion, active sensing, planning

## 1. INTRODUCTION

Multi-agent systems are becoming increasingly popular in target search and tracking. Many researchers proposed approaches for multiple agents to find an unknown number of dynamic or static targets from uncertain measurements (Chen and Dames, 2020b; Leonard and Zoubir, 2019; Chen and Dames, 2020a; Ramachandran et al., 2021; Ma et al., 2015; Zhou et al., 2019). Compared to a single-agent scenario, multi-agent systems (Claudio et al., 2018) improve detection by combining information from the agents. For instance, Liu et al. (2020) defined a target-search estimator that fuses measurements from several agents using the iterated-corrector hypothesis density filter (IC-PHD). Zhou et al. (2019) demonstrate resilient target tracking against worst-case sensor failures in a receding-horizon manner.

Most methods mentioned do not explicitly take environment exploration into account. In our earlier paper (Yousuf et al., 2022), we already proposed an exploration-based method for a single drone to explore a 2D environment in search of an unknown number of static targets at unknown positions.

The present paper extends the method of Yousuf et al. (2022), by adding *multiple* agents to explore a 2D environment. We consider both homogeneous and heterogeneous agents that explore the target space to find the unknown number of static targets, see Fig. 1. Homogeneous agents have identical capabilities (dynamics, sensors, etc), while heterogeneous agents differ in their dynamics or sensors.

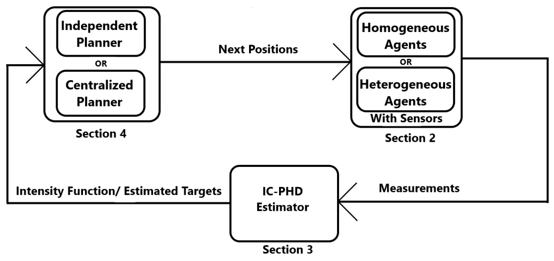


Fig. 1. Block diagram of our framework

The method for heterogeneous agents is motivated by our ongoing SeaClear project. In SeaClear, two robots, a remotely operated underwater vehicle (ROV) and an unmanned aerial vehicle (UAV), share a map and aim to find targets (litter) on the sea floor. Each robot is an agent. The UAV searches the area from the air, highlighting the regions with higher chances of containing targets, while the ROV investigates the highlighted regions to more accurately pinpoint the targets.

In our framework, each agent is equipped with a sensor with a limited field of view (FOV), which is modeled using a position-dependent probability of detection. For those targets that are detected, the sensor provides noisy measurements and an IC-PHD filter (Liu et al., 2020) that runs in the framework of Random Finite Sets (RFS) (Ma et al., 2015; Mahler, 2010; Chen et al., 2003; Vo et al., 2014; Charrow et al., 2014) is used to fuse information from all agents and to estimate the location of the targets. The IC-PHD filter generates estimated target positions in the form of an intensity function, a generalization of the probability density that integrates to the expected number of targets (Dames, 2020).

The key novelty of this paper consists of two multi-agent planning algorithms that generate the path of the agents by picking future discrete waypoints so as to maximize an objective function. In both algorithms, the objective func-

⋆ This work was been financially supported from H2020 SeaClear, a project that received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871295; and by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, SeaClear support project number PN-III-P3-3.6-H2020-2020-0060.

tion maximized by the planner consists of two components: target refinement and exploration. The target refinement component aims to better find the locations of already seen targets. The second, exploration component drives the agents towards unseen regions of the space to find new targets. The two different planner algorithms are: 1) independent control, where the agents explore the environment fully independently, and 2) centralized control, where the actions of the agents are chosen jointly. In contrast to the work of other researchers discussed above, this method — like our previous one in (Yousuf et al., 2022) — includes exploration.

We extensively test our framework in a series of simulations. As a baseline, we consider systematically generated lawnmower trajectories, as shown in Fig. 6. The proposed algorithms work better than this baseline to detect a large number of targets with either homogeneous or heterogeneous agents.

The remaining material is structured as follows, see also Fig. 1. Section 2 defines the problem, followed by the IC-PHD filtering framework in Section 3. In Section 4, we present the two different planners. Section 5 presents simulation results, and Section 6 concludes the paper.

## 2. PROBLEM FORMULATION

We consider $n_a$ agents that explore a 2D space (environment) $E$, to search for an unknown number of static targets, see Fig. 2, in as few steps as possible. We implement an RFS framework, as detailed in (Vo et al., 2005). The current position of the agent in Cartesian coordinates is denoted by $q_k^o$ at time step $k$, where $o = 1, 2, \ldots, n_a$ is the index of the agent.

At each step, agents receive a set of measurements about the targets, affected by noise. Agents are equipped with sensors that are capable of detecting targets depending on a FOV, as in Fig. 2. Note that the sensors of different agents may have different parameters. The probability of agent $o$ with position $q^o$ of detecting a target at position $x$ is given by $\pi_d^o(x, q^o) = Ge^{-\|\zeta^o\|/2}$, where $G \leq 1$ is a constant and $\zeta^o = (\frac{\mathcal{X}_x - \mathcal{X}_q^o}{\mathbb{F}_\mathcal{X}}, \frac{\mathcal{Y}_x - \mathcal{Y}_q^o}{\mathbb{F}_\mathcal{Y}})$ is the normalized distance of the target from the agents. Here, $(\mathcal{X}_x, \mathcal{Y}_x)$ denotes the target's position, $(\mathcal{X}_q^o, \mathcal{Y}_q^o)$ defines the agent's position, and $\mathbb{F}_\mathcal{X}$ and $\mathbb{F}_\mathcal{Y}$ are the width and length of the sensor field of view. Whether the target $x_{ik}$ is detected or not by agent $o$ is decided using a Bernoulli distribution, $b_{ik}^o \sim \text{ß}(\pi_d^o(x_{ik}, q_k^o))$. Then, the set of measurements $Z_k^o$ is defined as in (Gao et al., 2021):

$$Z_k^o = \left\{ \bigcup_{i \in \{1,\ldots,N_k\} \text{ s.t. } b_{ik}^o = 1} h_k^o(x) + \varrho_k^o \right\} \tag{1}$$

where

$$\begin{aligned} h_k^o &= [d_{ik}^o, \theta_{ik}^o]^T \\ d_{ik}^o &= \sqrt{(\mathcal{X}_{x_{ik}} - \mathcal{X}_{q_k}^o)^2 + (\mathcal{Y}_{x_{ik}} - \mathcal{Y}_{q_k}^o)^2} \\ \theta_{ik}^o &= \arctan \frac{\mathcal{Y}_{x_{ik}} - \mathcal{Y}_{q_k}^o}{\mathcal{X}_{x_{ik}} - \mathcal{X}_{q_k}^o} \end{aligned} \tag{2}$$

where $(\mathcal{X}_{x_{ik}}, \mathcal{Y}_{x_{ik}})$ is the position of target $i$ in the space $E$, and $d_{ik}^o, \theta_{ik}^o$ are the distance and bearing of target $i$ relative

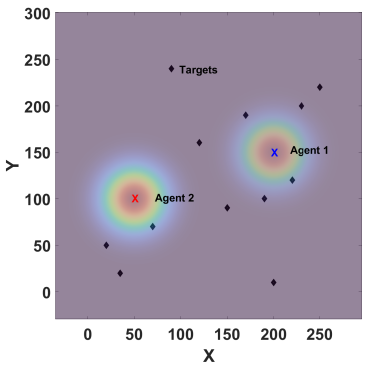


Fig. 2. 2D target space with 12 targets and 2 agents. The colors show the probability of observation of each agent at their current position, orange to blue means higher to lower probability.

to the agent's position. Moreover, $\varrho_{ik}^o \sim \mathcal{N}(., \mathbf{0}, R^o)$ is a Gaussian noise with mean $\mathbf{0} = [0, 0]^\top$ and covariance $R^o = \text{diag}[(\sigma^o)^2, (\sigma^o)^2]$.

The target measurement density $g^o(z_k|x_k)$ can be written as:

$$g^o(z_k^o|x_k) = \mathcal{N}(z_k^o, h_k^o(x), R^o) \tag{3}$$

i.e., $g^o(z_k^o|x_k)$ is a Gaussian density function, with covariance $R^o$ and centered on $h_k^o(x)$ from (2). This density will be used to estimate the target locations.

## 3. ESTIMATION FRAMEWORK

This section summarizes the Iterated-Corrector Probability Hypothesis Density (IC-PHD) filtering framework. In this framework, the objective is to estimate the location of the targets by fusing measurements from all agents. All computations are centralized. In Section 3.1, we discuss the intensity function and the single-agent PHD filter prediction and update operations, followed by IC-PHD update operations in Section 3.2. The elimination of targets found is defined in Section 3.3.

### *3.1 Single-agent PHD Filter*

The Probability Hypothesis Density (PHD) $D : E \to [0, \infty)$, or intensity function, is similar to a probability density function with the key difference that its integral over the entire domain is not 1, but the number of targets. The PHD filter performs Bayesian updates of an intensity function based on the target measurements and is summarized as:

$$\begin{aligned} D_{k|k-1} &= \Phi_{k|k-1}(D_{k-1|k-1}) \\ D_{k|k} &= \Psi_k(D_{k|k-1}) \end{aligned} \tag{4}$$

Here, $D_{k|k-1}$ is the prior intensity function predicted based on intensity function $D_{k-1|k-1}$ at time step $k-1$, and $D_{k|k}$ denotes the posterior generated after processing the measurements. The multi-target prior $D_{k|k-1}$ at step $k$ is defined by:

$$\begin{aligned} D_{k|k-1}(x_k) = \Phi_{k|k-1}(D_{k-1|k-1})(x_k) = \\ \Upsilon_k(x_k) + \int_{E^s} p_s(\xi)\delta_\xi(x_k) D_{k-1|k-1}(\xi) d\xi \end{aligned} \tag{5}$$

where $p_s(\xi)$ is the probability that the target still exists. In our specific problem, targets are stationary, so the

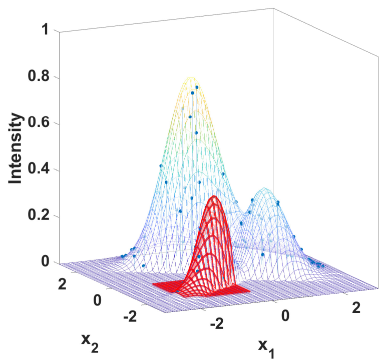


Fig. 3. Illustration of an intensity measure defined over 2D space

transition density of a target $x_k$ is defined as a Dirac delta $\delta_\xi(x_k)$, and $\Upsilon_k(x_k)$ denotes the intensity function of a new target appearing in the FOV, chosen here as a constant. The multi-target posterior $\Psi_k(D_{k|k-1})(x_k)$ can be written as:

$$D_{k|k}(x_k) = \Psi_k(D_{k|k-1})(x_k) =$$
$$\left[1 - \pi_{d_k}(x_k) + \sum_{z \in Z_k} \frac{\psi_{kz}(x_k)}{\langle \psi_{kz}, D_{k|k-1}\rangle (x_k)}\right] \cdot D_{k|k-1}(x_k) \tag{6}$$

where $\psi_{kz}(x_k) = \pi_d^o(x_k, q_k^o) g^o(z_k^o|x_k)$ denotes the overall probability density of detecting a target at $x_k$ with $g$ defined in (3). In practice, we employ the SMC-PHD filter (Vo et al., 2005), which uses a set of weighted particles to represent the intensity function. An example of an intensity function is given in Fig. 2, where the yellow color shows the peaks of the intensity function, and the circles represent the weighted particles. For example, the red patch in Fig. 2 is $D$ over region $S$, and $S$ is the corresponding rectangle lying in the $(x_1, x_2)$ plane. The integral of the intensity function over the red region gives the expected number of targets in $S$.

## 3.2 Sequential IC-PHD Filter

Let us next consider a set of agents $n_a$, receiving noisy measurements per our proposed sensor model. Each agent $o$ receives a measurement set $Z_k^o$ depending on the probability of detection $\pi_d^o(x, q_o)$, and computes likelihood function $g^o(z_k^o|x_k)$ based on those measurements. At time $k$, the IC-PHD filter (Liu et al., 2020) generates a posterior intensity function $D_{k|k}^o$ of each sensor $o$ in turn. In this filter, the prediction step is similar to that in (5). The prior is then taken to be the initial multi-agent posterior $D_{k|k}^0$, with index 0:

$$D_{k|k-1} = \Phi_{k|k-1}(D_{k-1|k-1})(x) =: D_{k|k}^0 \tag{7}$$

Then, the measurement sets of the agents will be processed sequentially, in the posterior updates:

$$\begin{aligned} D_{k|k}^o(x) &= \Psi_k^o(D_{k|k}^{o-1})(x), \; o = 1, \ldots, n_a \\ D_{k|k}(x) &:= D_{k|k}^{n_a}(x) \end{aligned} \tag{8}$$

Each IC-PHD posterior update equation is defined as:

$$D_{k|k}^o =$$
$$\left[1 - \pi_{d_k}^o(x_k) + \sum_{z_k^o \in Z_k^o} \frac{\psi_{kz}^o(x_k)}{\langle \psi_{kz}^o, D_{k|k-1}\rangle (x_k)}\right] \cdot D_{k|k}^{o-1} \tag{9}$$

for $o = 1, 2, \ldots, n_a$. The particle-based approximation from Section 3.1 is used. Note that for the special case of a single-sensor problem, the IC-PHD reduces to the original PHD filter.

## 3.3 Marking and Removal of Found Targets

The algorithm maintains two sets of targets (i) found targets and (ii) potential targets. At each step $k$, we extract the potential targets as clusters of particles with K-means. Then, two conditions are checked: (1) the width $W_i$ of each cluster of particles should be below threshold $\mathcal{T}_W$, and (2) the sum of the weights $\omega_j$ of the particles in the cluster should be above threshold $\mathcal{T}_m$. If these two conditions are satisfied, i.e., the cluster is very narrow and contains a high concentration of particles, we shift the corresponding target from potential to *found*.

To allow the drone to focus on refining poorly seen targets and finding new targets, we remove the particles corresponding to found targets, and to avoid reidentifying the found targets, we disregard future measurements likely to come from them, see (Yousuf et al., 2022) for details.

## 4. MULTI-AGENT PLANNER

Consider now the problem of designing a path for the agents to explore the 2D target space to quickly find the targets. Next states are picked by maximizing an objective function. The set of possible candidate next states of all agents is $\mathbf{Q}_k = \mathbb{Q}_k^1 \times \mathbb{Q}_k^2 \times \ldots \times \mathbb{Q}_k^{n_a}$, where $\mathbb{Q}_k^o$ is the set of candidate next states for agent $o$. $\mathbf{Q}_k$ is discrete and should be sufficiently rich for the agents to search for unknown targets, see e.g. Fig 4. We define two different cases: (i) independent planning, where each agent $o$ optimizes in its own set $\mathbb{Q}_k^o$, and (ii) centralized planning, where agents optimize together in the joint set of candidates $\mathbf{Q}_k$.

**Independent planner.** This planner works by making independent control decisions $q_{k+1}^{*o}$ for each agent. It is a direct application for each agent of the method in (Yousuf et al., 2022), where we used a single agent. Each agent runs their own PHD filter, like in equation (4), and takes decisions with:

$$q_{k+1}^{*o} = \operatorname{argmax}_{q_{k+1}^o \in \mathbb{Q}_k^o} \left\{\mathbb{T}_k^o(q_{k+1}^o) + \alpha \cdot \mathbb{E}^o(q_{k+1}^o)\right\} \tag{10}$$

The objective function has two components: target refinement $\mathbb{T}_k^o(q_{k+1}^o)$, aiming to better find the locations of the targets that were already seen; and exploration $\mathbb{E}^o(q_{k+1}^o)$, driving the agents to explore unseen regions. The tunable parameter $\alpha$ controls the tradeoff between the two components. The target refinement component is computed as the sum of the observation probabilities of particle cluster centers:

$$\mathbb{T}_k^o = \sum_{j=1}^{\mathbb{C}^o} \pi_d^o(\mathcal{C}_j^o, q_{k+1}^o) \tag{11}$$

and is different from the target refinement component of (Yousuf et al., 2022). In (11), $\mathbb{C}^o$ denotes the cluster count, and $\mathcal{C}_j^o$ is the center of the $j$th cluster. The center $\mathcal{C}_j^o$ has the meaning of an estimated target position. The intuition is that the probability of seeing estimated target locations is maximized.

An exploration function $\iota(x)$ is defined to compute $\mathbb{E}^o(q_{k+1}^o)$, initially set to 1 for the whole target space. At each step $k$, $\iota$ should decrease for each $x$ with an amount equal to the probability of detection $\pi_d^o(x, q_k^o)$. The meaning is that position $x$ has been explored to an amount equal to the probability of detection at that position. In practice, $\iota$ is implemented by interpolation on a 2D grid $x_{ij}$. Each grid point is initialized with:

$$\iota_0^o(x_{ij}) = 1, \forall i, j$$

then decreased with the rule:

$$\iota_k^o(x_{ij}) = \iota_{k-1}^o(x_{ij}) \cdot (1 - \pi_d^o(x_{ij}, q_k^o)), \forall i, j \tag{12}$$

Then, the exploration component of (10) is defined as:

$$\mathbb{E}^o(q_{k+1}^o) = \iota_k^o(q_{k+1}^o) \tag{13}$$

where $\iota_k^o$ is computed at $q_{k+1}^o$ using interpolation on the grid. It is important to observe that when no target has been detected (or all targets were marked as found, see Section 3.3), the agent's trajectory will be computed only based on the exploration component, and will fill the space with a lawnmower-like trajectory.

As mentioned earlier, each agent runs its own PHD filter (Section 3.1) to generate the particle clusters (estimated target positions) and applies the target removal method to found targets (Section 3.3). The center $\mathcal{C}_j^o$ of such a cluster is added to a set $\hat{X}$ of found targets that are common to all the agents. Agents therefore explicitly communicate with each other about target positions that have been found, and each agent will ignore measurements coming from any target in $\hat{X}$. Removal is done like in the last **for** loop of Algorithm 1.

**Centralized planner.** The centralized planner works by making a centralized control decision $\mathbf{q}_{k+1}^* = \left(q_{k+1}^{1*}, q_{k+1}^{2*}, \dots, q_{k+1}^{n_a*}\right)^T$ for all agents at once, ,with the following rule:

$$\mathbf{q}_{k+1}^* = \text{argmax}_{\mathbf{q}_{k+1} \in \mathbf{Q}_k} \left\{ \mathbf{T}_k(\mathbf{q}_{k+1}) + \alpha \cdot \mathbf{E}(\mathbf{q}_{k+1}) \right\} \tag{14}$$

In contrast to the independent planner, in this case, there is no separate set of weighted particle intensity function estimates for each agent. There is only one set of weighted particles generated using the IC-PHD filter based on the measurements of all the agents, as in Section 3.3. We extract clusters from this single set of particles, which are used to compute the target refinement component $\mathbf{T}_k(\mathbf{q}_{k+1})$ as follows.

$$\mathbf{T}_k(\mathbf{q}_{k+1}) = \mathcal{T}^1(q_{k+1}^1) + \mathcal{T}_k^2(q_{k+1}^2) \dots + \mathcal{T}_k^{n_a}(q_{k+1}^{n_a})$$

where $\mathcal{T}_k^o(q_{k+1}^o)$, with index $o = 1, 2, \dots, n_a$:

$$\mathcal{T}_k^o = \sum_{i=1}^{\mathbb{C}} \pi_d^o(\mathcal{C}_j, q_{k+1}^o) \tag{15}$$

Here, $\mathcal{C}_j$ is defined as the center of cluster $j$ in the set of particles, and $\mathbb{C}$ is the number of such clusters.

The exploration component $\mathbf{E}(\mathbf{q}_{k+1})$ is formulated as the product of the exploration components of each agent:

$$\mathbf{E}(\mathbf{q}_{k+1}) = \prod_{o=1}^{n_a} \iota_k^o(q_{k+1}^o) \tag{16}$$

Each exploration component is defined in the same way as for the independent planner. The intuition is that centralized exploration should cover more area compared to independent exploration, as agents are driven to explore different regions at each time step $k$.

After a target is marked as found, the center of the corresponding cluster is added to a set $\hat{X}$ of found targets. Similarly to the independent planner, after the target has been added, all particles related to it are deleted, and future measurements from that target are ignored. Algorithm 1 summarizes the centralized planner procedure.

**Algorithm 1:** Centralized control procedure at $k$

```
Generate set $\mathbf{Q}_k$ of candidate next states for all agents
Compute exploration component $\mathbf{E}$ using (16)
for each $\mathbf{q}_{k+1} \in \mathbf{Q}_k$ do
    compute target refinement $\mathbf{T}_k$ and exploration $\mathbf{E}$
Find best next joint state $\mathbf{q}_{k+1}^*$ using (14)
Execute K-means to find clusters $\mathcal{C}_j, j = 1, \dots, \mathbb{C}$
for each cluster $j = 1, \dots, \mathbb{C}$ do
    if $W_i \leq \mathcal{T}_W$ and $\sum_{j \in \mathbb{C}} \omega_j \geq \mathcal{T}_m$ then
        /* target found */
        delete all particles $i \in \mathcal{C}_j$
        $\hat{X} \leftarrow \hat{X} \cup \hat{x}_i$
Get measurements $Z_k^o$ from sensors of each agent $o$
For each agent $o$
for $H_k^o(x) \in Z_k^o$ do
    for $\hat{x} \in \hat{X}$ do
        $Z_{\text{aux}}^o = \{z_k^o \in Z_k^o \mid \|h^{-1}(z_k^o) - \hat{x}\| \leq \mathcal{T}_z\}$
        if $Z_{\text{aux}}^o$ is nonempty then
            /* remove measurement */
            $Z_k^o = Z_k^o \setminus \text{argmin}_{z_k^o \in Z_{\text{aux}}^o} \|h^{-1}(z_k^o) - \hat{x}\|$
Run filter from Section 3.2 with measurements $Z_k^o$
```

## 5. RESULTS

In this section, we present simulation results to validate the above framework. Section 5.1 presents results using homogeneous agents, followed by heterogeneous agents in Section 5.2. We compare how well the proposed algorithm performs versus a multi-agent lawnmower trajectory. These results also showcase the potential of our framework using heterogeneous agents e.g. for the SeaClear project. All simulations have been done using MATLAB.

### *5.1 Homogeneous-Agent Results*

In these experiments, we consider 2 agents, and 12 targets distributed uniformly in the 2D environment $E = [-30, 300]\text{m} \times [-30, 300]\text{m}$. The candidate position set of each agent has 8 different position choices: right, left, top, bottom, and the diagonals, as shown in Fig. 4 (left). 10 different runs are made. As a baseline, we use a lawnmower. The trajectory length is chosen as 120 steps for all experiments because the lawnmower needs this length to complete the search: at 120 steps, the agents' trajectories

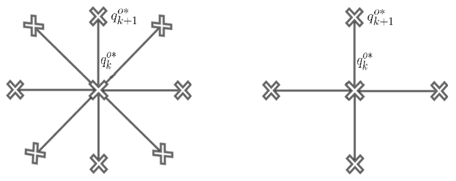


Fig. 4. Candidate next positions for one agent, where the current position is in the middle. Left: candidate set for homogeneous agents. Right: candidate set for heterogeneous agents

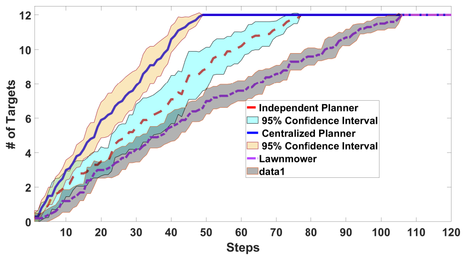


Fig. 5. Average number of targets detected across 10 maps with the 95% confidence intervals.

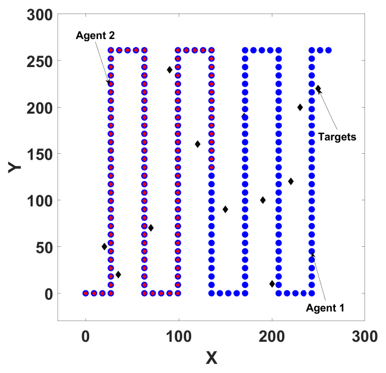
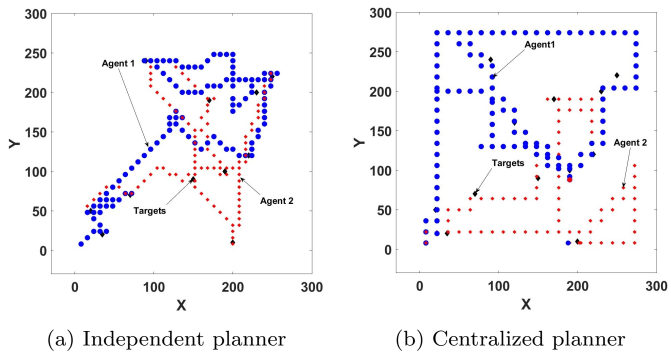


Fig. 6. Multi-agent trajectories with a lawnmower.



(a) Independent planner

(b) Centralized planner

Fig. 7. Examples of trajectories using the two new planners

meet in the middle, and they find all the targets. The threshold values in Algorithm 1 are experimentally set as $\mathcal{T}_W = 1.2$m, $\mathcal{T}_m = 2.5$m, and $\mathcal{T}_z = 5$m. The parameters of the probability of detection $\pi^o_d(x, q^o)$ are: $G = 0.98$, $\mathbb{F}_\mathcal{X} = \mathbb{F}_\mathcal{Y} = 25$. The field of view is the same for all experiments for fairness. The distance from the current position at $k$ to the future position at $k+1$ is set to 8 meters at each step. The starting position of the agents is set as [0,0], and it is the same in all runs.

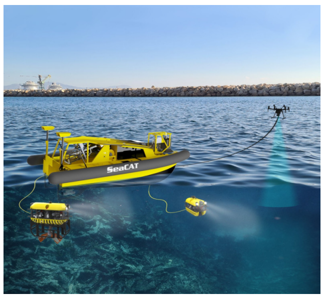
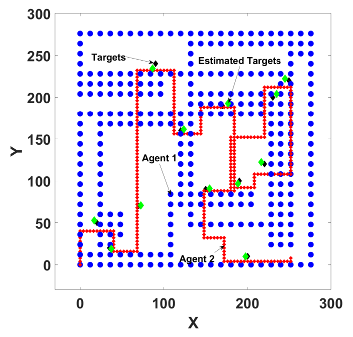


Fig. 8. SeaClear system concept

Fig. 5 shows the number of target detections over time, averaged in all the 10 experiments, along with their 95% confidence interval. The proposed algorithms find the target faster than the lawnmower. The centralized planner finds the targets faster than the independent planner because agents find targets at different locations. In Fig. 6 we show the lawnmower trajectories, where the agents start in opposite corners and meet in the middle. It is visible that the multi-agent lawnmower has covered the whole target space. Fig. 7 shows the agents' trajectories with the novel planners in one of the 10 experiments. The agents do not cover the whole environment uniformly but focus on relevant regions to find targets faster compared to the lawnmower. In Fig. 7a, for the independent planner, we see that since the agents take independent decisions, they sometimes focus on the same targets. In Fig. 7b, when the centralized planner is used to explore the target space, the agents focus on different targets.

## 5.2 *Heterogeneous-Agent Results*

To illustrate the performance of the proposed target localization algorithm in a more realistic scenario, we link our research problem to the ongoing project SeaClear `https://seaclear-project.eu/`. In SeaClear, we have two different robots: an unmanned aerial vehicle (UAV), and a remotely controlled underwater vehicle (ROV), both in search of litter on the sea floor, as shown in Fig. 8. The UAV is faster and has a larger sensor FOV. The ROV has better sensor accuracy but a lower speed and a smaller FOV. In the literature, to distinguish between target and sea life, Deep learning methods are used. Here we use the simulation with the simple sensor model of Section 2. We took agent 1 as the UAV and agent 2 as the ROV, to see how our algorithm handles these different types of agents. In this simulation, we used only the centralized planner, and the candidate set has 4 different position choices: right, left, top, and bottom, as shown in Fig. 4 (right). The initial states of both agents are set as [0; 0], in the 2D environment $E = [-30, 300]\text{m} \times [-30, 300]\text{m}$. The threshold values in Algorithm 1 are set as $\mathcal{T}_W = 1.2$m, $\mathcal{T}_m = 2.4$m, and $\mathcal{T}_z = 5$m. The parameters of the probability of detection for the first agent are $G = 0.98$, and the sensor FOV is set as $\mathbb{F}_\mathcal{X} = \mathbb{F}_\mathcal{Y} = 60$, and the distance from the current position at $k$ to the future position at $k+1$ is set to 12m. For the 2nd agent, $G = 0.98$, the sensor field of view is set as $\mathbb{F}_\mathcal{X} = \mathbb{F}_\mathcal{Y} = 15$, and the distance from the current position of agent 2 at step $k$ to the future position at $k+1$ is set to 4 m. The computation time is around 1.88s and 3.02s per time step for agents 1 and 2, respectively.



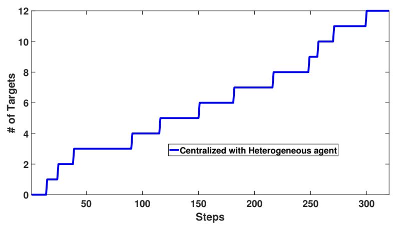


Fig. 9. Top: Heterogeneous agent trajectories using the centralized planner, along with the estimated target position. Bottom: Number of targets found over time.

In this simulation, we consider 12 targets uniformly distributed at random locations. The trajectory length is chosen as 320 steps since the slower agent takes more steps to find the targets. Fig. 9 (top) shows the agents' trajectories. It is visible that the first agent roughly solves a coverage problem: it explores the whole environment with a large FOV. The sensor of agent 1 is unable to detect the target well, so agent 1 only highlights those areas that have higher chances of containing targets. The second agent, having a smaller FOV and better accuracy, focuses on relevant regions highlighted by the faster agent, and finds all the targets. Fig. 9 (top) also shows the estimated target positions in green color. Fig. 9 (bottom) shows the number of target detections over time. The RMSE of the localization is 1.14 m. For a better understanding of this experiment, please refer to the full video of the heterogeneous agents' trajectories, available online at `http://rocon.utcluj.ro/files/multiagent_targetsearch.mp4`.

## 6. CONCLUSION

In this paper, we introduced a novel strategy that enables multiple agents to search for an unknown number of static targets in a 2D environment. These agents are equipped with imperfect sensors, which are noisy and may fail to detect targets within the field of view. To estimate the number and location of the targets, the iterated-corrector Probability Hypothesis Density (IC-PHD) filter is used. We proposed two novel planners with target refinement and exploration components, and validate the performance of our framework through simulations. Future work includes the deployment of the framework on the real robots of the SeaClear project, where challenges include identifying the sensor model and hardware integration.

## REFERENCES

Charrow, B., Michael, N., and Kumar, V.R. (2014). Active control strategies for discovering and localizing devices with range-only sensors. *Algorithmic Foundations of Robotics*, 107, 51–71.

Chen, J. and Dames, P. (2020a). Collision-free distributed multi-target tracking using teams of mobile robots with localization uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6968–6974. Las Vegas, NV, USA.

Chen, J. and Dames, P. (2020b). Distributed multi-target search and tracking using a coordinated team of ground and aerial robots. *Robotics science and systems*, 47, 971–977.

Chen, J., Xie, Z., and Dames, P. (2003). The semantic PHD filter for multi-class target tracking: From theory to practice. *Robotics and Autonomous Systems*, 149, 1–14.

Claudio, F., Ngu, V.B., Tuong, V.B., Giorgio, B., and Luigi, C. (2018). Robust fusion for multisensor multiobject tracking. *IEEE Signal Processing Letters*, 25(5), 640–644.

Dames, P. (2020). Distributed multi-target search and tracking using the PHD filter. *Autonomous robots*, 44, 673–689.

Gao, L., Battistelli, G., Chisci, L., and Farina, A. (2021). Fusion-based multidetection multitarget tracking with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 57(4), 2438–2458.

Leonard, M.R. and Zoubir, A.M. (2019). Multi-target tracking in distributed sensor networks using particle PHD filters. *IEEE Transactions on Signal Processing*, 159, 130–146.

Liu, L., Ji, H., Zhang, W., and Liao, G. (2020). Multi-sensor fusion for multi-target tracking using measurement division. *IET Radar Sonar Navigation*, 14, 1451–1461.

Ma, L., Xue, K., and Wang, P. (2015). Distributed multiagent control approach for multitarget tracking. *Mathematical Problems in Engineering*, 2015, 1–10.

Mahler, R. (2010). Multitarget Bayes filtering via first-order multi-target moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4), 1152–1178.

Ramachandran, Kumar, R., Fronda, Nicole, Sukhatme, and S, G. (2021). Resilience in multirobot multitarget tracking with unknown number of targets through reconfiguration. *IEEE Transactions on Control of Network Systems*, 8(2), 609–620.

Vo, B.N., S.Singh, and Doucet, A. (2005). Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4), 1224–1245.

Vo, B.N., Vo, B.T., and Phung, D. (2014). Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Transactions on Signal Processing*, 6(24), 6554–6567.

Yousuf, B., Lendek, Zs., and Buşoniu, L. (2022). Exploration-based search for an unknown number of targets using a UAV. In *Proceedings of 6th IFAC Conference on Intelligent Control and Automation Sciences,(ICONS 22), IFAC-Papers Online*, volume 55, 93–98. Cluj, Romania.

Zhou, L., Tzoumas, V., Pappas, G.J., and Tokekar, P. (2019). Resilient active target tracking with multiple robots. *IEEE Robotics and Automation Letters*, 4(1), 129–136.