Jointly-optimized Searching and Tracking with Random Finite Sets

Savvas Papaioannou, Panayiotis Kolios, Theocharis Theocharides,
Christos G. Panayiotou and Marios M. Polycarpou

Abstract—In this paper we investigate the problem of joint searching and tracking of multiple mobile targets by a group of mobile agents. The targets appear and disappear at random times inside a surveillance region and their positions are random and unknown. The agents have limited sensing range and receive noisy measurements from the targets. A decision and control problem arises, where the mode of operation (i.e. *search* or *track*) as well as the mobility control action for each agent, at each time instance, must be determined so that the collective goal of searching and tracking is achieved. We build our approach upon the theory of random finite sets (RFS) and we use Bayesian multi-object stochastic filtering to simultaneously estimate the time-varying number of targets and their states from a sequence of noisy measurements. We formulate the above problem as a non-linear binary program (NLBP) and show that it can be approximated by a genetic algorithm. Finally, to study the effectiveness and performance of the proposed approach we have conducted extensive simulation experiments.

Index Terms—Bayesian target tracking, Intelligent systems, Sensor control, Area coverage

NOMENCLATURE

$x_k \in \mathcal{X}$	Single target state vector at time k.
$z_k \in \mathcal{Z}$	Measurement vector at time k.
$s_k \in \mathcal{A}$	Single agent state vector at time k
$u_k \in \mathbb{U}_k$	Single agent control action at time k.
$X_k \in \mathcal{F}(\mathcal{X})$	Multi-target state (i.e. finite set) at time k.
$Z_k \in \mathcal{F}(\mathcal{Z})$	Measurement set (i.e. finite set) at time k.
$\pi_{k\|k-1}(x_k\|x_{k-1})$	Transitional density of x_k based on time $k-1$ given x_{k-1}.
$g_k(z_k\|x_k, s_k)$	Measurement likelihood function of z_k conditioned on x_k and s_k.
$p_D(x_k, s_k)$	Single agent sensing model i.e. probability of detection.
$p_{k\|k-1}(x_k\|z_{1:k-1})$	Predictive distribution of x_k based on time $k-1$ given $z_{1:k-1}$.
$p_k(x_k\|z_{1:k})$	Posterior distribution of x_k at time k given $z_{1:k}$.
$f_{k\|k-1}(X_k\|Z_{1:k-1})$	Multi-object predictive distribution of X_k based on time $k-1$ given $Z_{1:k-1}$.
$f_k(X_k\|Z_{1:k})$	Multi-object posterior distribution of X_k given $Z_{1:k}$.
$r_{k\|k-1}$	Single target probability of existence at time k based on time $k-1$ (prior).
r_k	Single target probability of existence at time k (posterior).
$v_{k\|k-1}$	Maximum number of hypothesized targets at time k based on time $k-1$.
v_k	Maximum number of hypothesized targets at time k (posterior).
$Z_{k\|k-1}$	Predicted ideal measurement set (PIMS) for time k based on time $k-1$.
$\hat{X}_{k\|k-1}$	Estimation of X_k based on time $k-1$.
$\hat{X}_k$	Estimation of X_k based on time k.
$\hat{n}_{k\|k-1}$	Estimation of the true number of targets n_k based on time $k-1$.
$\hat{n}_k$	Estimation of the true number of targets n_k based on time k.
σ_k	Variance of the estimated number of targets $\hat{n}_k$
$\tilde{\sigma}_k$	Normalized variance of the estimated number of targets $\hat{n}_k$
$S = \{1, 2, ..., \|S\|\}$	The finite set of all agents for searching and tracking.
ξ_{search}	Multi-agent searching objective function.
ξ_{track}	Multi-agent tracking objective function.

1 INTRODUCTION

The main objective of a search and rescue mission (e.g. ground search and rescue, air-sea rescue, etc) is to search for and provide aid to people who are in imminent danger as efficiently and safely as possible. When disasters happen e.g. earthquakes, marine disasters and aircraft accidents, the search and rescue (SAR) missions are of critical importance for finding survivors and saving lives. SAR missions however could be very dangerous and expensive (e.g. the search for the Malaysian Airlines Flight 370 that disappeared on 8 March 2014 had costed a total of more than US $150 million).

On the other hand, the miniaturization and cost reduction of electronic components and the recent advances in robotics and specifically in small unmanned aerial vehicles (UAVs) have spurred an unprecedented interest on intelligent mobile agents for group missions. Motivated by this, we believe that a team of autonomous mobile agents could become an important aid in many search and rescue missions by improving the efficiency while at the same time reducing the need to place the rescuers in dangerous

- *The authors are with the KIOS Research and Innovation Centre of Excellence (KIOS CoE) and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus.*
E-mail:{`papaioannou.savvas, pkolios, ttheocharides, christosp, mpolycar`}*@ucy.ac.cy*

situations. Take for instance a maritime disaster where a team of drones could be deployed to search for and track survivors until further assistance is available.

In this work we are interested in the task of searching for and tracking multiple survivors or targets of interest in SAR missions with a group of autonomous mobile agents. This is a challenging problem since a) the number of survivors that needs to be tracked at a time instance is random, unknown and thus needs to be estimated, b) the agents need to estimate the location of the survivors through noisy measurements and c) the mobile agents exhibit a limited sensing range thus there is a need for efficient searching. Finally, the problem becomes even more challenging due to the tradeoff which arises on how many resources to allocate in each task i.e. *searching* or *tracking* at each time instance so that the joint search and track (SAT) objective is achieved as best as possible.

To address the above challenges, we propose a unified probabilistic approach to jointly tackle the problem of searching and tracking of multiple moving targets by a team of mobile agents. The proposed approach, refereed to hereafter as JoSAT (Jointly-optimized Searching and Tracking) is formulated in the framework of recursive Bayesian multi-object stochastic estimation using random finite sets (RFS) [1], [2] which allows us a) to simultaneously estimate the time-varying number of targets and their states from a sequence of noisy measurements and b) avoids the problem of data-association in multi-target tracking. The contributions of this paper are:

- Provides a multi-agent probabilistic framework for jointly searching and tracking multiple targets inside a given surveillance area. We utilize the theory of random finite sets (RFS) in a multi-agent framework to accurately capture the inherent uncertainty present in many search-and-track (SAT) missions.
- Develops a novel decision (i.e. the agents can switch between *search* and *track* mode during the mission) and control (i.e. control the movement of all agents) algorithm which takes into account the stochasticity of the system in order to tackle the joint objective of searching and tracking.
- Formulates the decision and control problem as a non-linear binary optimization program which is then solved using a genetic algorithm.

The rest of the paper is organized as follows. Section 2 reviews the existing literature on searching and tracking by single and multiple agents, demonstrating in the way the contributions of this work. Section 3 outlines the problem addressed in this paper and Section 4 presents an overview of the proposed system. Section 5 provides a brief overview on the framework that the proposed approach is based on namely a) random finite sets (RFS) and b) Bayesian multi-object stochastic filtering and then Section 6 presents the details of the proposed approach. Section 7 reformulates the problem in the context of mathematical programming, demonstrates the complexity of the resulting non-linear binary program and elaborates on how the problem can be solved in practice using a genetic algorithm. Section 8 conducts an extensive performance analysis and finally, Section 9 concludes the paper and discusses future work.

2 Related Work

Coordinated teams of agents unlock significantly greater capabilities than what is possible by single-handed missions. By taking this into consideration, existing literature has looked into several challenging problems that multi-agent systems can successfully tackle.

The coverage problem is among the most cited problems, where a fleet of airborne agents need to spread across an area over the shortest time interval for situational awareness or when searching for particular targets. Research works have focused around task assignment, scheduling and path planning of the multiple agents, considering physical resource constraints such as the total number of agents, their battery levels and their communication ranges [3] [4] [5] [6] [7] [8].

Another challenging problem is that of localization and tracking of single or multiple targets [9] [10] [11]. For this problem both centralized and distributed solutions are being investigated for the envisioned tracking strategies which generally seek to minimize the tracking error for all detected targets [12] [13] [14].

Interestingly though, many of these problems are interconnected and interrelated and thus need to be looked at jointly especially in certain settings such as in search and rescue missions (SAR) where efficiency and effectiveness are of essence. The problem of SAR with single or multiple robots has attracted a wide interest in the recent years with the research community proposing methods from multi-robot cooperative learning [15] and path-planning strategies [16] to collaborative online task-planning by heterogenous robot teams [17] and autonomous multi-UAV systems for SAR missions [18]. Detailed surveys on multi-robot SAR can be found in [19], [20]. In this paper however, we are focusing on a particular sub-problem i.e. the problem of search and track which has also attracted a wide interest by the research community because of it importance in SAR missions. An interesting work in [21] develops a probabilistic approach for the SAT problem but only for the single-agent single-target case. Similarly, a more recent work [22] investigates the SAT problem for the single-agent single-target scenario, and proposes a model-predictive control (MPC) framework for the path-planning problem. The work [23] considers the cooperative management of groups of UAVs as an optimization problem of the information obtained through multi-target tracking. Compared to our work however, the work in [23], is mainly focused on improving the tracking accuracy without including the search objective. The works in [24], [25] develop an information-theoretic multi-agent SAT approach which is used to search for a single stationary target by minimizing the entropy of the target distribution at each time step. Moreover, [26] looked at the problem of route planning for multi-agent SAT missions and proposed a Fisher information based approach to route planning. However, in [26] the problems of false alarms and data association are not considered. Similarly, the work in [27] presents a receding horizon controller (RHC) approach to jointly drive a group of UAVs to SAT missions. In contrast, in the proposed approach the data-association problem can be avoided altogether. Notably, the classical approaches to multi-target tracking require to first solve the data association problem and then estimate the states of the targets. RFS-based approaches can estimate the states of the targets directly without requiring to first solve the data-association problem [28]. In addition, existing techniques make weak assumptions regarding the number of targets, the target dynamics and the measurement models.

Complementary to the aforementioned studies, the proposed system advances the state-of-the-art, by considering a multi-agent approach where the agents can dynamically switch their mode of operation between searching and tracking during the mission in order to jointly optimize the coupled objective of searching and multi-target tracking and by developing a unified probabilistic search-and-track framework based on random finite sets which avoids the problem of data-association and accounts for the uncertainties in the number of targets, target dynamics and measurement models. Finally, this work derives both optimal and heuristic solutions to solve the resulting joint searching and tracking problem.

3 Problem Definition

In this work we assume that the agents can operate in either *search* or *track* mode and that they can switch modes dynamically during the mission in order to optimize the joint search and track objective. The problem to solve can be stated as follows: *At each time step a fixed number of agents must find their mode of operation and the control actions which result in the best tradeoff between maximizing the search coverage and tracking all targets found in the surveillance area.*

Throughout this paper we consider the following modeling assumptions.

3.1 Single Target Dynamics

The single target state vector $x_k \in \mathcal{X}, k \in \mathbb{N}$ evolve in time according to the following equation:

$$x_k = \zeta_k(x_{k-1}) + \text{v}_k \tag{1}$$

where the $\mathcal{X} \subseteq R^{n_x}$ denotes the state space and the known function $\zeta_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ models the dynamical behavior of the target. Eqn. (1) describes the evolution of the state vector as a first order Markov process with transitional density $\pi_{k|k-1}(x_k|x_{k-1}) = p_\text{v}(x_k - \zeta_k(x_{k-1}))$. The random process $\text{v}_k \in \mathbb{R}^{n_x}$, which is referred to as the process noise, is IID according to the probability density function $p_\text{v}(.)$. The role of the process noise is to model random disturbances in the evolution of the state. Without loss of generality, in this paper we assume that the state vector $x_k \in \mathcal{X} \subseteq \mathbb{R}^4$ is composed of position and acceleration components i.e. $x_k = [p_x, \dot{p}_x, p_y, \dot{p}_y]^\top$ where (p_x, p_y) give the 2D position of the target in Cartesian coordinates and $(\dot{p}_x, \dot{p}_y)$ are the velocities of the target in the x and y direction respectively.

Suppose now that multiple targets x_k^i exists inside the surveillance region where we assume that the targets are independent of each other. The targets can spawn from anywhere in the state space $\mathcal{X}$ and target births and deaths occur at random times. This means that at each time k, there exist n_k target states $x_k^1, x_k^2, ..., x_k^{n_k}$, each taking values in the state space $\mathcal{X}$ where both the number of targets n_k and their individual states $x_k^i, \forall i$ are random and time-varying. The group of target states at time k can now be modeled as the set:

$$X_k = \{x_k^1, x_k^2, ..., x_k^{n_k}\} \in \mathcal{F}(\mathcal{X}) \tag{2}$$

where $\mathcal{F}(\mathcal{X})$ denotes the space of all finite subsets of $\mathcal{X}$. Note here that n_k is the true but unknown number of targets at time k which needs to be estimated along with the target states $x_k^i, \forall i$. In Sec. 5-6 it will become evident that X_k is actually a random finite set (RFS). We will refer to X_k as the *multi-target state*.

3.2 Single Agent Sensing Model

The ability of an agent to sense its 2D environment is modeled by the function $p_D(x_k, s_k)$ that measures the probability that a target with state x_k at time k is detected by an agent with state (i.e. 2D position coordinates) $s_k = [s_x, s_y]_k^\top \in \mathcal{A} \subseteq R^2$. More specifically the detection probability of a target with state x_k at position $p_k = Hx_k$ (where H is a matrix that extracts the xy-coordinates of a target from its state vector) from an agent with state s_k, is given by:

$$p_D(x_k, s_k) = \begin{cases} p_D^{\max} & \text{if } d_k < R_0 \\ \max\{0, p_D^{\max} - \eta(d_k - R_0)\} & \text{if } d_k \geq R_0 \end{cases} \tag{3}$$

where $d_k = \|Hx_k - s_k\|_2$ denotes the Euclidean distance between the agent and the target, $p_D^{\max}$ is the detection probability for targets that reside within R_0 distance (i.e. the agents primary radius) from the agent's position and finally η captures the reduced effectiveness of the agent to detect distant targets.

When an agent detects a single target it receives a measurement vector $z_k \in \mathcal{Z}$ which is related to the target state as follows:

$$z_k = h_k(x_k, s_k) + w_k \tag{4}$$

where $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$ denotes the measurement space and the function $h_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ projects the state vector to the measurement space. The random process $w_k \in \mathbb{R}^{n_z}$ is IID, independent of v_k and distributed according to $p_w(.)$. The probability density of measurement z_k for a target with state x_k when the agent is at state s_k is given by the measurement likelihood function $g_k(z_k|x_k, s_k) = p_w(z_k - h_k(x_k, s_k))$. Without loss of generality, in this paper we assume that the measurement vector $z_k \in \mathcal{Z}$ consists of range and bearing measurements.

That said an agent can receive m_k measurements $z_k^1, z_k^2, ..., z_k^{m_k}$ at each time step k each taking values in the measurement space $\mathcal{Z}$, where m_k and $z_k^i, \forall\, i$ are random and time varying. Thus the group of received measurements by an agent at time k can be modeled as the set:

$$Z_k = \{z_k^1, z_k^2, ..., z_k^{m_k}\} \in \mathcal{F}(\mathcal{Z}) \tag{5}$$

where $\mathcal{F}(\mathcal{Z})$ denotes the space of all finite subsets of $\mathcal{Z}$. In Sec. 5-6 it will be shown that Z_k is also a random finite set. Z_k will be referred to as the *multi-target measurement set*.

3.3 Agent Dynamics

Let $S = \{1, 2, ..., |S|\}$ be the set of all mobile agents that we have in our disposal operating in a discrete-time setting. At time k the 2D surveillance area $\mathcal{A} \subseteq \mathbb{R}^2$ is monitored by $|S|$ mobile agents with states $s_k^1, s_k^2, ..., s_k^{|S|}$, each taking values in $\mathcal{A}$. Each agent is subject to the following dynamics with bounded control inputs:

$$s_k^j = s_{k-1}^j + \begin{bmatrix} l_1 \Delta_R \cos(l_2 \Delta_\theta) \\ l_1 \Delta_R \sin(l_2 \Delta_\theta) \end{bmatrix}, \quad \begin{matrix} l_2 = 0, ..., N_\theta \\ l_1 = 0, ..., N_R \end{matrix} \tag{6}$$

where $s_{k-1}^j = [s_x^j, s_y^j]_{k-1}^\top$ denotes the position (i.e. xy-coordinates) of the j_th agent at time $k-1$, Δ_R is the radial

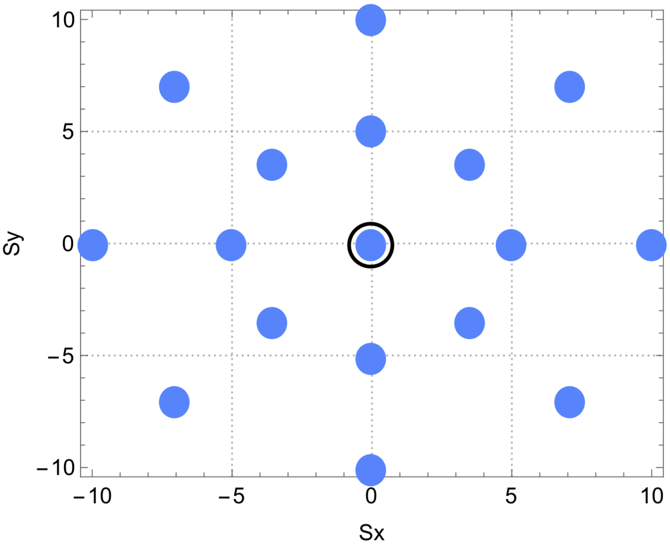

Fig. 1. The figure shows an illustrative example of the agent's admissible control actions $\mathbb{U}_k$ at time k. In this example the radial displacement $\Delta_R = 5$m, $N_R = 2$ and $N_\theta = 8$ which gives a total of 17 control actions, including the initial position of the agent at time $k-1$ which is marked with a circle.

step size, $\Delta_\theta = 2\pi/N_\theta$ and the parameters (N_θ, N_R) control the number of possible control actions. We denote the set of all admissible control actions of agent j at time k as $\mathbb{U}^j_k = \{s^{j,1}_k, s^{j,2}_k, ..., s^{j,|\mathbb{U}_k|}_k\}$ as computed by Eqn. (6). An illustrative example is shown in Fig. 1.

3.4 JoSAT Objective

The jointly optimized search and track objective can now be more formally defined as:

$$(\mathrm{P}_1)\underset{u^j_k, \forall j \in S}{\arg\min} \left[\xi^\alpha_{\text{search}}(\{u^j_k : j \in \alpha\}) + \sum_{\substack{j=1 \\ j \notin \alpha}}^{|S|} \xi^j_{\text{track}}(u^j_k, Z^j_{k|k-1,u^j_k}) \right] \tag{7}$$

$$\begin{aligned} \text{subject to} \quad & \alpha \in \mathbb{P}(S) \\ & u^j_k \in \mathbb{U}^j_k \ , \forall j \in S \\ & Z^j_{k|k-1,u^j_k} = Z_{\text{PIMS}}(\hat{X}^j_{k|k-1}, u^j_k) \ , \forall j \in S \\ & \left\| u^i_k - u^j_k \right\| > d_{\min} \ , \forall\ i \neq j \end{aligned}$$

where α is the decision variable for the agents' operating mode and $\mathbb{P}(S) = \Big\{\emptyset, \{1\}, \{2\}, \ldots, \{|S|\}, \{1,2\}, \ldots\Big\}$ denotes the power set of S. The function $\xi^\alpha_{\text{search}}(\{u^j_k : j \in \alpha\})$ takes as input a set of control actions for the agents in search mode and returns the total cost of searching and the function $\xi^j_{\text{track}}(u^j_k, Z^j_{k|k-1,u^j_k})$ returns the cost of tracking when the control u^j_k is taken and the predicted multi-target measurement set $Z^j_{k|k-1,u^j_k}$ is supposed to have been received by agent j. $\hat{X}^j_{k|k-1}$ denotes the estimated multi-target state for time k from agent j based on time $k-1$.

The cost of tracking depends on the control u^j_k that is applied to agent j through the future received measurements (i.e. depending on the control action taken, different measurements will be received). The measurement generator function $Z_{\text{PIMS}}(\hat{X}^j_{k|k-1}, u^j_k)$ generates the predicted ideal multi-target measurement set based on $\hat{X}^j_{k|k-1}$ and u^j_k.

Finally, the constraint $\left\| u^i_k - u^j_k \right\| > d_{\min}, \forall\ i \neq j$ does not allow any two agents to get close to each other. This is by design since in this work there is no benefit of two agents to be close to each other, and for instance search the same area or perform tracking of the same target.

We should note here that the optimization of Eqn. (7) finds the joint control actions over all agents, and at the same time partitions the agents into two groups i.e. search or track so that the joint search-and-track cost is minimum. In other words, at each time-step, the system automatically assigns the agents to searching or tracking mode and selects the joint control actions which minimize the JoSAT objective. Also note, that in this work we do not assume any time delays between agents i.e. all agents have access to the same clock and they are fully synchronized.

4 System overview

In this section we give an overview of the proposed system architecture, shown in Fig. 2, and we outline how we have incorporated the theory of multi-object filtering to the problem of multi-agent joint searching and tracking.

To summarize, an agent $j \in S$ can operate in either *search* or *track* mode and in each mode the agent is subject to a set of admissible control actions denoted by $\mathbb{U}^j_k$ for time k. The agents can dynamically switch between the two modes during the mission in order to maximize the joint global objective of searching and tracking i.e. Eqn (7). When a subset of agents $\alpha \subseteq S$ is in *search* mode the collective objective over all agents $j \in \alpha$ is the maximization of the covered area. In other words the agents should traverse the surveillance area in such a way so that the coverage is maximized. On the other hand when a subset of agents $\alpha^\mathrm{C} \subseteq S$ (where α^C denotes the complement of α with respect to S) is in *track* mode, their objective is the maximization of the tracking performance, i.e. to move in locations where the number of targets and their states is estimated as accurately as possible. Deciding on the subset of agents in *search* or *track* mode is governed by the respective objective functions that control the movement of the agents to achieve the best tradeoff. In other words the proposed controller solves the optimization of Eqn. (7).

In order to achieve this, the proposed system takes the following steps: The number of targets and their states at time k are jointly modeled as a random finite set X_k (also known as *multi-target state*) which is a generalization of a random variable/vector (see Sec. 5).

Each agent maintains a *multi-object* probability distribution (see Sec. 5-6) which jointly accounts for the uncertainty in the number of targets and their states (i.e. position and velocity) inside its sensing range (this multi-object probability distribution is a generalization of the probability density function on random finite sets i.e. X). Let the posterior *multi-object* distribution at time $k-1$ be denoted by f^j_{k-1} for agent j. To be precise $f^j_{k-1} = f^j_{k-1}(X^j_{k-1}|Z^j_{1:k-1})$ where X^j_{k-1} is conditioned on the received measurement sets $Z^j_{1:k-1}$ up to time $k-1$, however we will often use the short notation i.e. f^j_{k-1} to denote this distribution.

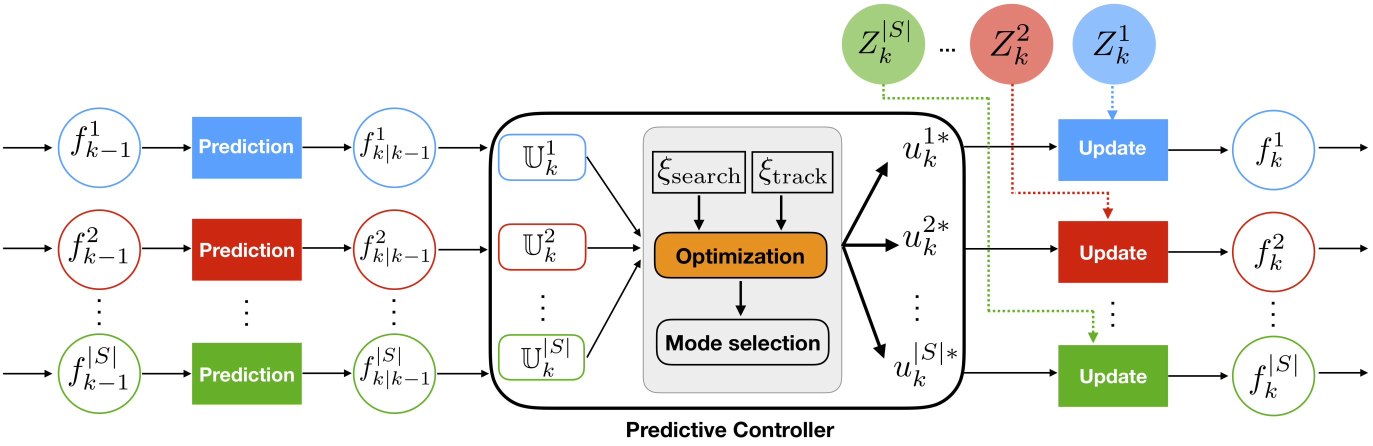

Fig. 2. The figure illustrates the proposed system architecture with $S = \{1, 2, ..., |S|\}$ agents. The following recursion is shown: a) for each agent j the predictive multi-object probability distribution $f^j_{k|k-1}$ is first computed from the posterior f^j_{k-1} using the prediction step, b) the estimated numbed of targets and their states are extracted from each predictive multi-object distribution, c) an optimization problem is then solved where different hypothetical control actions from the set of admissible control actions $\mathbb{U}^j_k$ are generated for each agent. The mode of operation i.e. *search* or *track* and the corresponding control actions for all agents are jointly selected so that the total cost of searching (given by the searching objective function i.e. ξ_{search}) and tracking (given by the tracking objective function i.e. ξ_{track}) is minimized. d) Finally, the agents move to the new states according to the selected optimized controls $\{u^{1*}_k, \dots, u^{|S|*}_k\}$ and compute the posterior multi-object distribution f^j_k using the received measurements Z^j_k at time k. The same procedure is the repeated for the next time instance.

In the first step the predictive density $f^j_{k|k-1} = f^j_{k|k-1}(X^j_k|Z^j_{1:k-1})$ is obtained through a prediction step as the figure shows. This allows us to make predictions regarding the number of targets and their states for the next time-step. The estimated number of targets $(\hat{n}^j_{k|k-1})$ and their states $(\hat{X}^j_{k|k-1})$ for the next time-step are extracted from the predictive density.

Based on these predictions the proposed controller finds the combination of control actions $(u^j_k \in \mathbb{U}^j_k \; \forall j)$ as well as the mode of each agent which minimizes the convex sum of the search and track objective functions (see Sec. 6).

The optimal control actions $u^{1*}_k, ..., u^{|S|*}_k$ are then applied to the agents. Each agent then moves to its new state at time k where it receives the multi-target measurement set Z^j_k (if it exists). This measurement set at time k is then used in the update step i.e. each agent uses the received measurement set to compute the posterior distribution $f^j_k = f^j_k(X^j_k|Z^j_{1:k})$. The updated (i.e. final) estimates of the number of targets $\hat{n}^j_k$ and their states $\hat{X}^j_k$ for time k are then computed from f^j_k for each agent j. The total estimated number of targets at time k is then computed as $\hat{n}_k = \sum_j \hat{n}^j_k, \forall j$ and their states as $\hat{X}_k = \bigcup_j \hat{X}^j_k, \forall j$. This procedure is repeated recursively over time.

We should note here that the number of targets (and their states) is being estimated recursively over time. Based on these estimates, the mode of operation should be carefully chosen. For instance when no targets are inside the surveillance area, all agents should be in *search* mode, etc. It is also worth noting that the control actions taken, affect the received measurements which in turn affect the estimation of the target state during the update step. Hence, to optimize the control actions would require the knowledge of all the future measurements. In the next section we provide an overview of random finite sets for multiple target tracking which we use subsequently to address the aforementioned challenge in our proposed approach.

5 Background on Multi-Target Tracking

Multiple target tracking (MTT) [29], [30] refers to the problem of estimating the number and states of multiple targets from noisy sensor measurements in the presence of false alarms or clutter. MTT is nowadays found in many applications including surveillance, defense, autonomous vehicles, robotics, etc. In this section we only give a brief overview on the various MTT techniques. A more detailed description of MTT algorithms can be found in [29], [31].

In this work we divide MTT algorithms in two main categories namely data association-based and data association free. Data association-based MTT algorithms such as the MHT, JPDA and GNN [29], [31] require to first solve the measurements-to-tracks assignment problem and only then proceed with the multi-target state estimation. On the other hand, data association free methods such as the RFS-based PHD, CPHD and Multi-Bernoulli filters [28], [30] can bypass the data association problem and proceed directly with the multi-target state estimation. Depending on the application, this property could be highly desirable since it significantly reduces the computational complexity of MTT algorithms.

The PHD filter [32] is the first computationally desirable approximation of the multi-target Bayes posterior. In particular it propagates the first-order statistical moment of the multi-target posterior in place of the full posterior distribution. The computationally more expensive CPHD filter [33] provides a better performance compared to the PHD filter since it jointly propagates the first-statistical moment and the cardinality distribution. Unlike the PHD and CPHD filters which propagate moments the multi-Bernoulli filter [34] approximates the true multi-target posterior distribution as multi-Bernoulli and propagates the parameters of a multi-Bernoulli distribution. The multi-Bernoulli filter avoids many of the problems present in the PHD/CPHD filters i.e. these filters require a clustering step to extract state estimates which introduces an additional source of error and increases the computationally complexity. This makes the multi-Bernoulli filter a more attractive solution with linear complexity in the number of targets and measurements.

We should also point out that with the introduction of labeled RFSs [35], [36] and in particular with the inception of the generalized labeled multi-Bernoulli (GLMB) density the multi-target Bayes filter can tackle the data-association problem and provide target tracks. More specifically, the GLMB multi-target filter [37] is able to simultaneously estimate the number of targets and their states from a set of noisy observations in the presence of data association uncertainty, detection uncertainty and clutter. However, compared to the multi-Bernoulli filter the computational complexity of the GLMB filter is at best cubic in the number of measurements. An extension to the multi-Bernoulli filter i.e. the Labeled multi-Bernoulli (LMB) filter, was presented in [38]. The LMB filter achieves better performance compared to the multi-Bernoulli filter while at the same time outputs target tracks. This however comes with an increased computational cost i.e. its computational complexity is at worst cubic in the number of measurements. Recently the authors in [39] have presented a more efficient implementation of the GLMB filter with linear complexity in the number of measurements and quadratic in the number of targets by combining the prediction and update into a single step.

5.1 Random Finite Sets

A random finite set (RFS) is a finite-set-valued random variable, which differs from a random vector in two ways: a) the number of elements in a RFS is random and b) the order of the elements in a RFS is irrelevant. More specifically, a RFS X is completely specified by a) its cardinality distribution $\rho(n) = p(|X| = n),\ n \in \mathbb{N}_0$ which defines a probability distribution over the number of elements in X and b) by a family of conditional joint symmetric probability distributions $p(x_1, ..., x_n|n),\ \ x_1, ..., x_n \in \mathcal{X}$ that characterize the distribution of its elements over the state space $\mathcal{X}$. In connection with our problem, the random variable n can be used to model the time-varying number of targets (i.e. the number of survivors at some point in time) and thus $\rho(n)$ is the probability distribution of the number of targets.

The belief density or otherwise known as the *multi-object* probability density function (pdf) $f(X)$ of the RFS X is given by: $f(X) = f(\{x_1, ..., x_n\}) = n!\rho(n)p(x_1, ..., x_n|n)$ and the notion of integration is given by the set-integral which is defined as:

$$\int_{\mathcal{F}(\mathcal{X})} f(X)\delta X = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{x_1, ..., x_n\}) d_{x_1} ... d_{x_n} \tag{8}$$

where $\mathcal{F}(\mathcal{X})$ is the space of finite subsets of $\mathcal{X}$. The following RFSs are relevant in this paper:

5.1.1 Bernoulli RFS

The Bernoulli RFS X can either be empty with probability $1 - r,\ r \in (0, 1)$ or be a singleton set (i.e. its set cardinality is equal to one) with (existence) probability r and with its element distributed over the state space $\mathcal{X}$ according to pdf $p(x)$. The Bernoulli multi-object pdf is given by:

$$f(X) = \begin{cases} 1 - r & \text{if } X = \emptyset \\ rp(x) & \text{if } X = \{x\} \end{cases} \tag{9}$$

Thus a Bernoulli RFS can be completely characterized by the parameter set $(r, p(x))$.

5.1.2 Multi-Bernoulli RFS

The multi-Bernoulli RFS $X = \bigcup_{i=1}^{v} X_i$ is a union of a fixed number v (where v is known) of independent Bernoulli RFSs X_i with parameters $\{(r^i, p^i(.))\}_{i=1}^{v}$ with multi-object pdf given by (see [30] pp. 102):

$$f(\{x_1, ..., x_n\}) = \sum_{1 \leq i_1 \neq, ..., \neq i_n \leq v} \Big(Q_{i_1, ..., i_n} {\cdot} p^{i_1}(x_1) {\cdots} p^{i_n}(x_n) \Big) \tag{10}$$

where

$$Q_{i_1, ..., i_n} = \left(\prod_{i=1}^{v} (1 - r^i) \right) \cdot \frac{r^{i_1}}{1 - r^{i_1}} \cdots \frac{r^{i_n}}{1 - r^{i_n}}$$

Here the notation $\sum_{1 \leq i_1 \neq, ..., \neq i_n \leq v}(.)$ enumerates the n-permutations of v i.e. $P^v_n = \frac{v!}{(v-n)!}$. In other words, it enumerates all ordered arrangements of n elements taken from a set of size v. A multi-Bernoulli distribution is completely specified with the parameter set $\{(r^i, p^i(.))\}_{i=1}^{v}$. In connection with our problem v can be used to model the maximum number of hypothetical targets maintained by an agent.

5.1.3 Poisson RFS

The poisson RFS X has a cardinality distribution which is Poisson with parameter λ i.e. $\rho(n) = \frac{e^{-\lambda}\lambda^n}{n!},\ n = 0, 1, 2...$ and elements which are independent and identically distributed (IID) random variables and distributed according to $p(x)$ on $\mathcal{X}$. The multi-object pdf is given by:

$$f(X) = e^{-\lambda} \prod_{x \in X} \kappa(x) \tag{11}$$

where $\kappa(x) = \lambda\ p(x)$ is called the intensity function for the Poison RFS.

5.2 Multi-object Stochastic Filtering

In stochastic filtering [40] we are interested in the posterior probability density $p_k(x_k|z_{1:k})$ of some hidden state x_k at time k given all measurements $z_{1:k} = z_1, ..., z_k$. Assuming an initial density on the state $p_0(x_0)$, the posterior density at time k can be computed using the Bayes recursion as:

$$p_{k|k-1}(x_k|z_{1:k-1}) = \int \pi_{k|k-1}(x_k|x_{k-1})\ p_{k-1}(x_{k-1}|z_{1:k-1})\ dx_{k-1} \tag{12}$$

$$p_k(x_k|z_{1:k}) = \frac{g_k(z_k|x_k)\ p_{k|k-1}(x_k|z_{1:k-1})}{\int g_k(z_k|x_k)\ p_{k|k-1}(x_k|z_{1:k-1})\ dx_k} \tag{13}$$

where Eqn. (12) and (13) are referred to as the prediction and update steps respectively. The function $\pi_{k|k-1}(x_k|x_{k-1})$ models the uncertainty of the current state given the previous state and is referred to as the transitional density and the function $g_k(z_k|x_k)$ describes the distribution of measurements given the current state and is referred to as the measurement likelihood function. At each time step the hidden state is usually extracted from the posterior distribution using the expected a posteriori (EAP) or the maximum a posteriori (MAP) estimators.

The above recursion can be extended [28] in the case of RFSs. Suppose now that the hidden state and measurements are RFS and that at time $k - 1$ the posterior multi-object pdf of X is given by $f_{k-1}(X_{k-1}|Z_{1:k-1})$. The predicted and updated multi-object densities are then given by the multi-object Bayes recursion:

$$f_{k|k-1}(X_k|Z_{1:k-1}) \\ = \int \phi_{k|k-1}(X_k|X_{k-1})\ f_{k-1}(X_{k-1}|Z_{1:k-1})\ \delta X_{k-1} \tag{14}$$

$$f_k(X_k|Z_{1:k}) = \frac{\varphi_k(Z_k|X_k)\ f_{k|k-1}(X_k|Z_{1:k-1})}{\int \varphi_k(Z_k|X_k)\ f_{k|k-1}(X_k|Z_{1:k-1})\ \delta X_k} \tag{15}$$

where the integrals in Eqn. (14)-(15) are set integrals, and the functions $\phi_{k|k-1}(X_k|X_{k-1})$ and $\varphi_k(Z_k|X_k)$ are the multi-object transitional density and the multi-object likelihood function respectively. Because the recursion above has no analytic solution in general, several approximations exist (as in [30]) which propagate only summary statistics instead of the full posterior density. A prominent approach is the multi-Bernoulli filter [34], [41] which is used in this work and elaborated in Sec. 6.

6 JoSAT Multi-Target Framework

This section develops the components of the JoSAT system using random finite sets and derives the multi-agent objective functions for the tasks of searching and tracking. Intuitively, the collection of targets at each time instance can be seen as a set which changes size as targets appear and disappear from the surveillance region (e.g. due to births and deaths). Since this set consists of a random number of random variables it can be modeled as a random finite set and the same is true for the received measurements as we detail below.

6.1 Multi-target Dynamics

Earlier in subsection 3.1 we have described the single target dynamics, in this subsection we will use random finite sets to describe the multi-target dynamics.

Using the RFS theory we can now define a multi-object dynamic model i.e. the evolution in time of the RFS X_k as follows:

$$X_k = \left[\bigcup_{x_{k-1} \in X_{k-1}} \Psi(x_{k-1}) \right] \cup \Gamma_k \tag{16}$$

where $\Psi(x_{k-1})$ is a Bernoulli RFS which models the evolution of the set from the previous state, with parameters $(p_S(x_{k-1}), \pi_{k|k-1}(x_k|x_{k-1}))$. Thus a target with state x_{k-1} continues to exists at time k with surviving probability $p_S(x_{k-1})$ and moves to a new state x_k with transition probability $\pi_{k|k-1}(x_k|x_{k-1})$, (see Eqn. (1)). Otherwise the target dies with probability $1 - p_S(x_{k-1})$. Note here that X_k is a multi-Bernoulli RFS assuming that the RFSs constituting the union in Eqn. (16) are mutually independent. Finally, the term Γ_k denotes the multi-Bernoulli RFS of spontaneous births with uniform spatial distribution over $\mathcal{X}$. We now have a multi-object transition model i.e. Eqn. (16) which jointly incorporates motion, birth and death for multiple targets.

6.2 Multi-target Measurement Model

At each time step an agent can receive a time-varying number of measurements i.e. the number of measurements is random and depends on: a) the number of targets inside the agent's sensing range which is random as well, b) whether all targets have been detected or not, and c) whether measurements come from targets or from clutter (i.e. false alarms). That said, the measurements received at some time instance can be modeled as a random finite set. The multi-target measurement set Z^j_k of agent j at time k is given by:

$$Z^j_k = \left[\bigcup_{x^j_k \in X^j_k} \Theta(x^j_k) \right] \cup \mathrm{K}_k \tag{17}$$

where $X^j_k \subseteq X_k$ is the multi-target state perceived by the j_{th} agent. $\Theta(x^j_k)$ is a Bernoulli RFS which models the target generated measurements with parameters $(p_D(x^j_k, s^j_k), g_k(z^{ji}_k|x^j_k, s^j_k))$. Thus a target with state x^j_k at time k is detected by the j_{th} agent with state s^j_k with probability $p_D(x^j_k, s^j_k)$ (see Eqn. (3)) and generates a measurement z^{ji}_k with likelihood $g_k(z^{ji}_k|x^j_k, s^j_k)$ (see Eqn. (4)) or is missed with probability $1 - p_D(x^j_k, s^j_k)$ and generates no measurements. Additionally an agent can receive false alarms measurements i.e. the term K_k is a Poisson RFS which models the set of false alarms or clutter received by an agent at time k with intensity function $\kappa_k(z^{ji}_k) = \lambda f_c(z^{ji}_k)$, where in this paper $f_c(.)$ denotes the uniform distribution over $\mathcal{Z}$.

Note here that Eqn. (17) accounts for the detection uncertainty and clutter and that in this work the following holds by design $X_k = \bigcup_j X^j_k, \forall j$ and $X^{j_1}_k \cap X^{j_2}_k = \emptyset,\ \forall j_1, j_2$ i.e. a target cannot be tracked by more than one agent.

6.3 Tracking Multiple Targets

When the agent j is in *track* mode the objective is to estimate at each time k both the number of targets n^j_k and their respective states $x^{ji}_k, \forall i \in \{1, ..., n^j_k\}$ inside its sensing range from a sequence of noisy measurement sets $Z^j_{1:k}$. In essence we would like to compute the multi-object density of X^j_k given the measurement sets $Z^j_{1:k}$ i.e. $f^j_k(X^j_k|Z^j_{1:k})$ using the multi-object Bayes recursion of Eqn. (14) - (15). The RFS formulation of subsections 6.1-6.2 however, allows us to compute this recursion with a multi-Bernoulli filter [34] which approximates the multi-object posterior density as multi-Bernoulli distribution and propagates only its parameters $\{(r^{ji}_k, p^{ji}_k(.))\}_{i=1}^{v^j_k}$ in time instead of the full distribution. For brevity, in the rest of the paper, we will denote the multi-Bernoulli distribution at time k as $f_k = \{(r^i_k, p^i_k(.))\}_{i=1}^{v_k}$.

Furthermore, the multi-Bernoulli filter recursion is summarized here. A detailed description of the multi-Bernoulli filter can be found in [34]. The multi-Bernoulli filter proceeds as follows: The predictive multi-object density at time k is given by:

$$f_{k|k-1} = f^{\text{persist}}_{k|k-1} \cup f^{\text{birth}}_k \tag{18}$$

where $f^{\text{persist}}_{k|k-1} = \{(r^i_{k|k-1}, p^i_{k|k-1}(.))\}_{i=1}^{v_{k-1}}$ describes the set of persisting targets from the previous time-step and $f^{\text{birth}}_k = \{(r^i_k, p^i_k(.))\}_{i=1}^{\Gamma_k}$ are the parameters of a multi-Bernoulli RFS for the new-born targets at time k. $f^{\text{persist}}_{k|k-1}$ is further given by:

$$r^i_{k|k-1} = r^i_{k-1}\,\langle p^i_{k-1}, p_S \rangle \tag{19}$$

$$p^i_{k|k-1}(x) = \frac{\langle \pi_{k|k-1}(x|.), p^i_{k-1}(.) p_S(.) \rangle}{\langle p^i_{k-1}, p_S \rangle} \tag{20}$$

where the notation $< f, g >$ is defined as $\int f(x)g(x)dx$.

Let the predictive multi-object density denoted by $f_{k|k-1} = \{(r^i_{k|k-1}, p^i_{k|k-1}(.))\}_{i=1}^{v_{k|k-1}}$ where now the term $v_{k|k-1}$ denotes the number of hypothesized Bernoulli components after accounting for the target births and let the multi-object measurement set at time k be $Z_k = \{z^1_k, ..., z^{m_k}_k\}$. The posterior multi-object distribution at time k is given by a multi-Bernoulli distribution with parameters:

$$f_k = f^{\text{legacy}}_k \cup f^{\text{meas}}_k \tag{21}$$

where $f^{\text{legacy}}_k = \{(r^i_k, p^i_k(.))\}_{i=1}^{v_{k|k-1}}$ contains legacy tracks i.e. updates of predicted tracks, assuming that no measurements were collected from them and given by:

$$r^i_k = r^i_{k|k-1} \frac{1 - \left\langle p^i_{k|k-1}, p_D(., s_k) \right\rangle}{1 - r^i_{k|k-1} \left\langle p^i_{k|k-1}, p_D(., s_k) \right\rangle} \tag{22}$$

$$p^i_k(x_k) = p^i_{k|k-1}(x_k) \frac{1 - p_D(x_k, s_k)}{1 - \left\langle p^i_{k|k-1}, p_D(., s_k) \right\rangle} \tag{23}$$

where $p_D(x_k, s_k)$ is the probability of detecting a target as described in subsection 3.2. $f^{\text{meas}}_k = \{(r_k(z), p_k(., z))\}_{z \in Z_k}$ contains new tracks i.e. joint updates of all predicted tracks using each of the measurements separately; and given by:

$$r_k(z) = \frac{\sum_{i=1}^{v_{k|k-1}} \frac{r^i_{k|k-1}(1-r^i_{k|k-1})\left\langle p^i_{k|k-1}, L^z_k \right\rangle}{\left(1-r^i_{k|k-1}\left\langle p^i_{k|k-1}, p_D(., s_k)\right\rangle\right)^2}}{\kappa_k(z) + \sum_{i=1}^{v_{k|k-1}} \frac{r^i_{k|k-1}\left\langle p^i_{k|k-1}, L^z_k \right\rangle}{1-r^i_{k|k-1}\left\langle p^i_{k|k-1}, p_D(., s_k)\right\rangle}} \tag{24}$$

$$p_k(x_k, z) = \frac{\sum_{i=1}^{v_{k|k-1}} \frac{r^i_{k|k-1}}{1-r^i_{k|k-1}} p^i_{k|k-1} L^z_k(x_k)}{\sum_{i=1}^{v_{k|k-1}} \frac{r^i_{k|k-1}}{1-r^i_{k|k-1}} \left\langle p^i_{k|k-1} \cdot L^z_k \right\rangle} \tag{25}$$

where $L^z_k(x_k) = g_k(z_k|x_k) p_D(x_k, s_k)$.

The posterior multi-object density at time k for agent j is thus given by $f^j_k = \{(r^{ji}_k, p^{ji}_k(.))\}_{i=1}^{v^j_k}$ from where the estimated number of targets $\hat{n}^j_k$ and the multi-target state $\hat{X}^j_k$ can be computed (see Sec. 6.4). The total estimated number of targets at time k in the surveillance area is then computed as $\hat{n}_k = \sum_j \hat{n}^j_k, \forall j$ and the multi-target state which accounts for all targets in the area as $\hat{X}_k = \bigcup_j \hat{X}^j_k, \forall j$.

6.4 Multi-Agent Tracking Objective

Let the multi-Bernoulli distribution $f^j_k = \{(r^{ji}_k, p^{ji}_k(.))\}_{i=1}^{v^j_k}$ from agent j where v^j_k is the maximum number of hypothesized targets, r^{ji}_k denotes the probability of existence of the i_{th} hypothesized target (i.e. how likely the i_{th} hypothesized target is a true target) and $p^{ji}_k(.)$ denotes the corresponding posterior density of the state of the i_{th} target.

The expected a posteriori (EAP) estimate of number of targets $\hat{n}^j_k$ can be computed as $\hat{n}^j_k = round(\sum_i r^{ji}_k)$ and the multi-target state $\hat{X}^j_k$ can then be computed by taking the $\hat{n}^j_k$ components with the highest probabilities of existence and extracting the single target states (i.e. $\hat{x}_k$) from their individual posterior densities as $\hat{X}^j_k = \bigcup_{i=1}^{\hat{n}^j_k} \{\hat{x}_k = \arg\max\ p^{ji}_k(x_k)\}$.

In tracking, we wish to estimate as accurately as possible the $\hat{n}^j_k$ and $\hat{X}^j_k$ for all j agents. Intuitively, one way to achieve this is to minimize the variance in the posterior distribution f^j_k from which we derive the above estimates. In this work we build upon this and we denote the tracking objective function we wish to minimize by $\xi^j_{\text{track}}(u^j_k, Z^j_k),\ u^j_k \in \mathbb{U}^j_k$ if we were going to apply at time k the control action u^j_k and subsequently observe at the same time-step the measurement set Z^j_k. Note here that the received measurement set Z^j_k depends on u^j_k (i.e. the control action taken) by the agent j through Eqn. (3) - (4). Also observe that the objective function we wish to minimize depends on an unknown future measurement set, i.e. the measurement set Z^j_k will be received once the control u^j_k has been determined and applied. In other words, this objective function can be evaluated after the control action u^j_k has been taken, because only then the measurement Z^j_k becomes available.

A common approach [42], [43] around this problem is to take the statistical expectation of the objective function with respect to all possible values of measurements i.e. $\mathbb{E}[\xi^j_{\text{track}}(u^j_k, Z^j_k)]$ thus the optimization problem becomes $u^{j\star}_k = \arg\min\ \mathbb{E}\left[\xi^j_{\text{track}}(u^j_k, Z^j_k)\right]$. However, computing this expectation is computationally intensive because it requires the generation of an ensemble of measurement sets (pseudo-measurements) Z^j_k for each hypothesized control action.

An alternative and computationally cheaper approach which we adopt in the paper uses the predicted ideal measurement set (PIMS) [44] i.e. the noise-free clutter-free measurement set which is most likely to be obtained when a particular control action is applied. We denote the PIMS measurement set as $Z^j_{k|k-1}$ and thus the control problem now becomes:

$$u^{j\star}_k = \underset{u^j_k \in \mathbb{U}^j_k}{\arg\min}\ \xi^j_{\text{track}}(u^j_k, Z^j_{k|k-1}) \tag{26}$$

To optimize Eqn. (26) let the multi-Bernoulli distribution at time $k-1$ be given by $f^j_{k-1} = \{(r^{ji}_{k-1}(u^j_{k-1}), p^{ji}_{k-1}(., u^j_{k-1}))\}_{i=1}^{v^j_{k-1}}$ where now by using this notation we explicitly show its dependence on the control action that was taken for the time-step $k-1$. At time k the agent has available a set of admissible control actions $u^j_k \in \mathbb{U}^j_k$ given by Eqn. (6), one of which should be selected in order to take the agent to the next state at time k. First we compute the multi-Bernoulli predictive density $f^j_{k|k-1} = \{(r^{ji}_{k|k-1}(u^j_{k-1}), p^{ji}_{k|k-1}(., u^j_{k-1}))\}_{i=1}^{v^j_{k|k-1}}$ which is obtained without yet performing any control action. Then we apply a pre-estimation step where the number of targets $\hat{n}^j_{k|k-1}$ is first estimated from the predictive density by counting the Bernoulli components which exhibit a probability of existence $r^{ji}_{k|k-1} > 0.5, \forall i \in \{1...v^j_{k|k-1}\}$. The states of those targets are then extracted from their spatial distributions to obtain $\hat{X}^j_{k|k-1}$. Then for a given hypothesized control action $u^j_k \in \mathbb{U}^j_k$ the corresponding PIMS $Z^j_{k|k-1,u^j_k}$ is generated as:

$$\begin{aligned} Z^j_{k|k-1,u^j_k} = Z^j_{k|k-1,u^j_k} \cup \\ \{\underset{z}{\arg\max}\ g_k(z|\hat{x}^j_{k|k-1}, u^j_k)\},\ \forall \hat{x}^j_{k|k-1} \in \hat{X}^j_{k|k-1} \end{aligned} \tag{27}$$

Suppose now that for each pair $(u^j_k, Z^j_{k|k-1,u^j_k})$ a pseudo-update is performed using Eqn. (22) - (25) to compute the (pseudo) posterior density $\hat{f}^j_k(X^j_k|Z^j_{k|k-1,u^j_k}, u^j_k)$ from where the number of targets is estimated. We consider the variance of this estimate as a measure of uncertainty which should be minimized. More specifically let $\hat{f}^j_k = \{(r^{ji}_k(u^j_k, Z^j_{k|k-1,u^j_k}), p^{ji}_k(., u^j_k, Z^j_{k|k-1,u^j_k}))\}_{i=1}^{v^j_{k,u^j_k}}$ then:

$$\hat{n}^j_k(u^j_k, Z^j_{k|k-1,u^j_k}) = \sum_{i=1}^{v^j_{k,u^j_k}} r^{ji}_k(u^j_k, Z^j_{k|k-1,u^j_k}) \tag{28}$$

$$\sigma^j_k(u^j_k, Z^j_{k|k-1,u^j_k}) = \sum_{i=1}^{v^j_{k,u^j_k}} r^{ji}_k(u^j_k, Z^j_{k|k-1,u^j_k}) \times (1 - r^{ji}_k(u^j_k, Z^j_{k|k-1,u^j_k})) \tag{29}$$

where σ^j_k denotes the variance of the number of targets estimate. In essence, by minimizing the cardinality variance σ^j_k agent j chooses the control action which results in the best estimate regarding the number of targets. This variance is maximized when $r^{ji}_k(u^j_k, Z^j_{k|k-1,u^j_k}) = 0.5, \forall i$ thus we can define the normalized variance as $\tilde{\sigma}^j_k = 4\sigma^j_k/v^j_k$, where we have dropped the dependence on the control action and measurements for notational clarity.

Using the normalized variance we define the single agent objective function to be minimized as:

$$\xi^j_{\text{track}}(u^j_k, Z^j_{k|k-1,u^j_k}) = (\tilde{\sigma}^j_k - 1)\sqrt{\frac{\hat{n}^j_k}{\mathcal{V}_{\text{cap}}}} + 1 \tag{30}$$

where $\mathcal{V}_{\text{cap}}$ is a known design constant which specifies the maximum number of targets that our agent is able to track at a given time which is referred to as the agent tracking capacity in this paper. The above objective function is bounded in the closed interval $[0..1]$ and it reaches the minimum cost (i.e. best) when the agent reaches its maximum tracking capacity i.e. $\hat{n}^j_k = \mathcal{V}_{\text{cap}}$ with the maximum accuracy (i.e. $\tilde{\sigma}^j_k = 0$). On the other hand the agent receives the maximum cost when it tracks targets with the worst accuracy (i.e. $\tilde{\sigma}^j_k = 1$). It is not desirable to perform poorly in tracking mode thus the agent will try to find alternative controls for which the number of targets tracked is maximized and/or the variance is minimized. Additionally, an agent might even switch to search mode if this results in better collective search-and-track performance, rather than inaccurately tracking targets. It is worth noting that the agent will receive the maximum cost in the case where the number of targets being tracked is 0 i.e. $\hat{n}^j_k = 0$. This behavior is by design since in this situation we wish to switch the agent to *search* mode.

Finally we define the multi-agent objective function for the tracking task as the average of their individual objective costs:

$$\frac{1}{|\alpha|}\sum_{j\in\alpha}\xi^j_{\text{track}}(u^j_k, Z^j_{k|k-1,u^j_k}) \tag{31}$$

where $\alpha \subseteq S$ denotes the set of agents in *tracking* mode. As before we wish to minimize the objective function of Eqn. (31). In other words given a set of agents $\alpha \subseteq S$ we would like to drive the agents to the states which result in the minimization of Eqn. (31) or the maximization of the tracking accuracy.

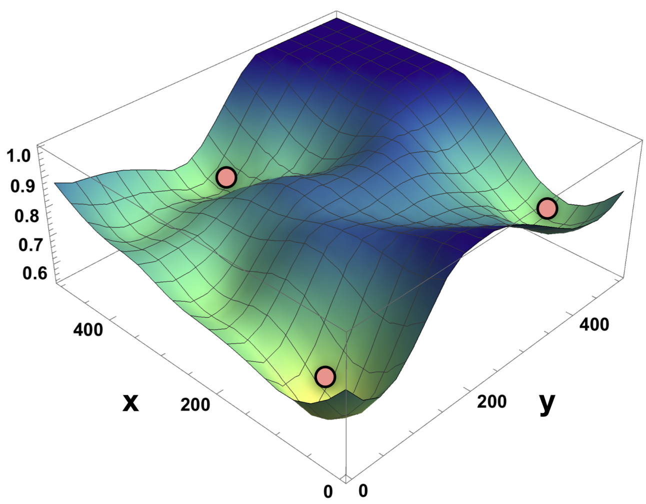

Fig. 3. The figure shows an illustrative example of the search value (Z axis) i.e. Eqn. (33) over a surveillance region of size 500m by 500m for 3 agents. The position of agents are marked with red circles. In this figure dark areas indicate high search values i.e. in these regions targets are likely to not be detected due to the limited sensing range of the agents. Thus the agents have incentive to move towards these regions.

6.5 Multi-Agent Search Objective

In the previous sub-section we have defined the objective function for the task of tracking for single and multiple agents. In this section we are going to do the same for the task of searching. Let us define the *search value* at time k for specific location $p = (p_x, p_y) \in \mathcal{A}$ for the agent j just after control action $u^j_k \in \mathbb{U}^j_k$ has been applied as:

$$\xi^j(p, u^j_k) = 1 - p_D(\tilde{p}, u^j_k) \tag{32}$$

where the function $p_D(\tilde{p}, u^j_k)$ is the agent's sensing model and $\tilde{p} = [p_x, 0, p_y, 0]$ constructs a location vector compatible with Eqn. (3). The idea of defining the search value as the complement of the probability of detection reflects the fact that locations with low probability of detection, should have increased value for searching. As a consequence distant locations with respect to the agent's location appear to have increased search value i.e. these areas are worth exploring in order to find new targets. Assuming pairwise detection independence between all agents and all targets, the search value at location p when accounting for a set of agents $\alpha \subseteq S$ is given by:

$$\xi^\alpha(p, \{u^j_k : j \in \alpha\}) = \prod_{j\in\alpha} \xi^j(p, u^j_k) \tag{33}$$

where $u^j_k \in \mathbb{U}^j_k$. This is shown in Fig. (3). Finally, we define the total search value over the surveillance area, i.e. the multi-agent search objective function as:

$$\xi^\alpha_{\text{search}}(\{u^j_k : j \in \alpha\}) = \frac{1}{A}\int_{\mathcal{A}} \xi^\alpha(p, \{u^j_k : j \in \alpha\})dp \tag{34}$$

where $\mathcal{A} \subset \mathbb{R}^2$ and A is the total area of the 2D surveillance region. We should note here that the above coverage problem does not consider the current or future target locations whatsoever. However, it will be very interesting to investigate in the future a multi-agent coverage problem which takes into account locations with high probability of target appearance.

7 Mathematical Programming Reformulation

To solve the proposed problem, we derive hereafter a mathematical program formulation (P$_1$) which takes into account the mode of operation and control actions in order to jointly optimize the search and track objective. Let $j \in \{1, \dots, |S|\}$ represent all available agents and $i \in \{1, \dots, |\mathbb{U}_k|\}$ the control actions for each agent. The following formulation considers all possible control actions i.e., $u_k^j \in \mathbb{U}_k^j$, as described in subsection 3.3. The binary variable $a_{ji} \in \{0, 1\}$ identifies which control is taken by agent j when in search mode, and binary variable $b_{ji} \in \{0, 1\}$ indicates the control decision for agent j when in tracking mode. Using this notation, P$_1$ can be transformed to the following binary nonlinear program:

$$(\mathrm{P}_2) \min w\xi_{\text{search}}(a) + (1-w)\xi_{\text{track}}(b) \tag{35}$$

$$\text{s.t.} \sum_{i=1}^{\mathbb{U}} a_{ji} + \sum_{i=1}^{\mathbb{U}} b_{ji} = 1, \ \ \forall j \in |\mathcal{S}| \tag{36}$$

$$\left\| \lambda_1 u_k^{ji} + \lambda_2 u_k^{j'i} \right\| > d_{\text{min}} \lambda_1 \lambda_2 \tag{37}$$

$$\lambda_1 \in \{a_{ji}, b_{ji}\}, \lambda_2 \in \{a_{j'i}, b_{j'i}\}, \ j \neq j'$$

$$a_{ji} \in \{0,1\}, b_{ji} \in \{0,1\}, \tag{38}$$

$$\forall j, j' \in \{1, \dots, |S|\}, i \in \{1, \dots, |\mathbb{U}_k|\}$$

The objective function, as expressed in eq. (35), is a convex sum (using scalar value w) of the search cost and the track cost achieved for a particular task assignment (i.e. either searching or tracking) and control decision (i.e. which control action to take next). The w value is a user-defined parameter representing the level of emphasis given to the task of searching versus the task of tracking. The linear equality constraints in eq. (36) ensure that each agent is only in the searching or tracking mode at any instance in time and only a single control decision in made in that time instance. Finally, eq. (37) ensures that all agents are a minimum distance d_{min} apart at any point in time which is achieved by computing the distance for any control points (i.e., u_k^{ji}) multiplied by the decision made in any of the 2 modes of operation (searching or tracking) indicated by variables $a_{ji}, \ b_{ji}$ using indicators $\lambda_1, \ \lambda_2$. Specifically, variables $\lambda_1, \ \lambda_2$ check every possible combination of a_{ji} and b_{ji} values against all other $a_{j'i}$ and $b_{j'i}$ values to ensure that agents in either search or track mode are not closer that a minimum distance d_{min} from each other.

Similarly to (P$_1$), the searching function can be expressed as follows:

$$\xi_{\text{search}}(a) = \frac{1}{A} \int_{\mathcal{A}} \prod_{j=1}^{|S|} \prod_{i=1}^{|\mathbb{U}|} \left(1 - p_D^i(\tilde{p}, u_k^{ji} a_{ji})\right) dp \tag{39}$$

which is a product of the probabilities that no agent detects a target at a specific location $\tilde{p}$ in the field. These probabilities are then integrated over the whole field $\mathcal{A}$ to compute the total search value of the particular combination of agents and specific control decisions. As shown in Eqn. (39) this search value is normalized by $\frac{1}{A}$ where A is the maximum value attained when all probabilities equal to 0. The tracking accuracy can be expressed as follows:

$$\xi_{\text{track}}(b) = \frac{1}{\tau} \sum_{j=1}^{|S|} \sum_{i=1}^{|\mathbb{U}|} c_{ji} b_{ji} \tag{40}$$

which is a sum of the costs $c_{ji} = \xi_{\text{track}}^j(u_k^{ji}, Z_{k|k-1, u_k^{ji}}^j)$ of an agent j being in tracking mode taking control action i, when control decision $b_{ji} = 1$. The tracking value can be normalized by $\tau = \sum_{j=1}^{|S|} \sum_{i=1}^{|\mathbb{U}|} b_{ji}$ which is the total number of agents in track mode.

Clearly, $\xi_{\text{search}}(a)$ jointly computes the search cost for a subset of agents being in search mode by the multiplication shown in Eqn. (39) that produce non-linear outputs. Coupled with the binary variables of the discrete control decisions results to an NP-hard optimization problem that is difficult to solve exactly in practice [45]. Nevertheless, for the particular structure of (P$_2$) and the binary encoding of the decision variables, a genetic algorithm (GA) can be applied to provide adequate solutions in practice. GA is a search-based technique that follows natural selection principles of solutions found to be relatively good [46]. To do that, a population of candidate solutions are generated over the search space and evaluated using the objective function of the problem (fitness function). The fittest solutions are then recombined and mutated to generate new populations for the next iteration. The process terminates when specific conditions are met, including the objective function improvement over consecutive iterations or at a maximum number of iterations.

A genetic algorithm is used to solve (P$_2$) with the following key elements:

Algorithm 1 : Proposed decision and control algorithm

```
Input: $f_{k-1}^j, \ \forall j \in \mathcal{S}$
Calculate the predictive multi-object density $f_{k|k-1}^j$ using Eqn. (18)-(20).
Estimate the number of targets from $f_{k|k-1}^j$ as $\hat{n}_{k|k-1}^j = \sum_{i=1}^{v_{k|k-1}^j} \mathbf{1}_{(r_{k|k-1}^{ji} > 0.5)}$ and extract the corresponding multi-target state $\hat{X}_{k|k-1}^j$.
repeat
    iter++
    Generate a candidate GA population of hypothesized control actions $u_k^{ji} \in \mathbb{U}_k^j, \forall j$ using Eqn. (6) with constraints (36) - (37).
    According to the generated hypothetical controls generate PIMS as $Z_{k|k-1}^j = Z_{k|k-1}^j \cup \{\arg\max_z g_k(z|\hat{x}_{k|k-1}^j, u_k^{ji} b^{ji})\} \ \ \forall \ \hat{x}_{k|k-1}^j \in \hat{X}_{k|k-1}^j$.
    Perform the pseudo-update step using PIMS to calculate the posterior multi-object density $\hat{f}_k^j, \forall j$ using Eqn. (21)-(25).
    Let $\Lambda_{\text{iter}} = w\xi_{\text{search}}(a) + (1-w)\xi_{\text{track}}(b)$ according to (35)
    Evaluate $\Lambda_{\text{iter}}$ and select the fittest solutions according to (36) - (38)
until (iter < iter$_{max}$) and ($\Lambda_{\text{iter}} - (\Lambda_{\text{iter}} - 1) < \epsilon$)
return optimal controls $\{u_k^{j*}\}, \forall j \in \mathcal{S}$
Apply optimal control action $u_k^{j*}$.
Receive the actual measurement set $Z_k^j$.
Compute posterior multi-object density $f_k^j$ using Eqn. (21)-(25).
Estimate $\hat{n}_k^j$ and $\hat{X}_k^j$ from $f_k^j$ (as in Sec. 6.4) , $\forall j$.
Estimate total number of targets $\hat{n}_k = \sum_j \hat{n}_k^j, \forall j$ and their states as $\hat{X}_k = \bigcup_j \hat{X}_k^j, \forall j$.
```

1) Binary decision variables $\{a_{ji}, b_{ji}\}, \forall j \in S,\ i \in \{1, \dots, |\mathbb{U}_k|\}$
2) The fitness function is reflected by the objective function in $(P2)$, i.e., Eqn. (35), and so are the linear equality constrains in Eqn. (36).
3) Control decisions that result to agents coming at a distance closer to d_{min} are prune to satisfy constraint Eqn. (37).
4) Candidate solutions are generated around controls of the same mode of each agent.
5) The algorithm terminates after a maximum number of iterations has been reached or when the fitness function did not improve more than ϵ.

The complete algorithm of the proposed system is shown in Algorithm (1). The algorithm implements the recursion which is shown in Fig. (2) and discussed throughout this paper.

8 Evaluation

8.1 Implementation Details

In order to evaluate the performance of the proposed approach we have conducted several numerical experiments involving various number of agents and targets for a variety of scenarios. In each experiment we compare the proposed approach with the ground-truth either qualitatively or quantitatively. In this section we include an extensive set of the conducted numerical simulations.

More specifically, we assume that the targets maneuver in an area of 500m × 500m and that the single target state at time k is described by $x_k = [p_x, \dot{p}_x, p_y, \dot{p}_y]_k^\top$ where (p_x, p_y) give the position of the target in Cartesian coordinates and $(\dot{p}_x, \dot{p}_y)$ are the velocities of the target in the x and y direction respectively. For the target dynamics in Eqn. (1), we consider that each target is moving according to the near constant velocity model with the process noise being Gaussian. Thus the single target transitional density is given by $\pi_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; Fx_{k-1}, Q)$ where:

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix},\ Q = \begin{bmatrix} T/3 & T/2 & 0 & 0 \\ T/2 & T & 0 & 0 \\ 0 & 0 & T/3 & T/2 \\ 0 & 0 & T/2 & T \end{bmatrix}$$

with sampling interval $T = 1$s. Note that we have used a linear motion model in our evaluation for the target dynamics and so Eqn. (1) becomes $x_k = Fx_{k-1} + \text{v}_k$. The target survival probability from time $k-1$ to time k is constant and does not depend on the target's state i.e. $p_S(x_{k-1}) = 0.99$.

Once the agent detects a target it receives bearing and range measurements thus the measurement model of Eqn. (4) is given by:

$$h_k(x_k, s_k) = \left[\arctan\left(\frac{s_y - p_y}{s_x - p_x}\right),\ \|s_k - Hx_k\|_2 \right]$$

where $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. The single target likelihood function is then given by $g_k(z_k|x_k, s_k) = \mathcal{N}(z_k; h_k(x_k, s_k), \Sigma^\top \Sigma)$ and sigma is defined as $\Sigma = \text{diag}(\sigma_\phi, \sigma_\zeta)$. The standard deviations $(\sigma_\phi, \sigma_\zeta)$ are range dependent and given by:

$$\sigma_\phi = \phi_0 + \beta_\phi \|s_k - Hx_k\|_2$$
$$\sigma_\zeta = \zeta_0 + \beta_\zeta \|s_k - Hx_k\|_2^2$$

with $\phi_0 = 2\pi/180$ rad, $\beta_\phi = 10^{-5}$ rad/m, $\zeta_0 = 1$ m, and $\beta_\zeta = 5 \times 10^{-5}$ m^{-1}. False alarm measurements (i.e. clutter) are generated with a Poisson rate $\lambda_k = 10$ uniformly distributed over the measurement space. The agent's sensing model parameters take the following values $p_D^{\text{max}} = 0.99$, $\eta = 23 \times 10^{-4}$ and $R_0 = 30$m. The agent's dynamical model is shown in Fig. 1 where the radial displacement $\Delta_R = 5$m, $N_R = 2$ and $N_\theta = 8$ which gives a total of 17 control actions, including the initial position of the agent. The tuning parameter w which controls the weight given to searching and tracking is set to 0.5 unless otherwise specified. Finally, the parameter $\mathcal{V}_{\text{cap}}$ is set depending on the simulation scenario to the maximum number of targets that we wish a single agent to be able to track. The genetic algorithm was implemented using the *ga* Matlab function. We should note here that any optimization/meta-heuristic method which can handle the binary program of Eqn. (35) with integer constraints, can be used in place of the genetic algorithm. However the genetic algorithm fits best to the structure of our problem. Finally, in order to handle the non-linear measurement model we have implemented a Sequential Monte Carlo (SMC) version [34], [47] of the multi-Bernoulli filter. When the target dynamics and the measurement model are linear and the probability of detecting targets is state independent the Gaussian version of the multi-Bernoulli filter provides a computationally more efficient solution [34]. Table 1 shows a list of variables and common values used in the experimental evaluation of the proposed approach.

TABLE 1
List of variables used for simulations

Symbol	Description	Values, [Units]
$\mathcal{A}$	Surveillance area	500 × 500, [m]
x_k	Target state	$[p_x, \dot{p}_x, p_y, \dot{p}_y]_k^\top$, [m, m/s, m, m/s]
T	Sampling interval	1, [s]
p_S	Target probability of survival	0.99
σ_ϕ	Bearing meas. noise (std)	varies, [rad]
σ_ζ	Range meas. noise (std)	varies, [m]
λ_k	False alarms rate	10 [meas. per time-step]
p_D^{max}	Max prob. of detection	0.99
η	Sensing model parameter	23×10^{-4}
R_0	Sensing model parameter	30 [m]
Δ_R, N_R, N_θ	Agent motion model params.	5 [m], 2, 8
GACT	GA Constraint Tolerance	10^{-6}
GAFT (ϵ)	GA Function Tolerance	10^{-4}
GAPS	GA Population Size	400
GAMG	GA Max Generations	$150|\mathbb{U}||S|$
w	Eqn. (35) mode weight	0.5

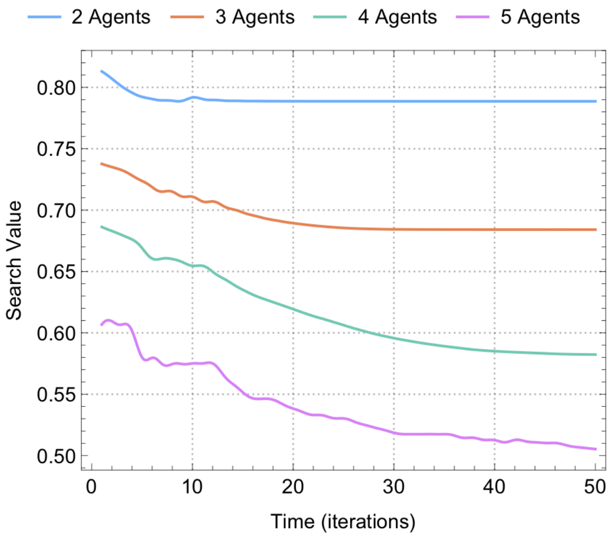

Fig. 4. The figure shows the evolution of the search cost for different number of agents inside a surveillance area of size 500m by 500m. As we can observe the search cost is decreasing over time which implies that the area coverage is increasing.

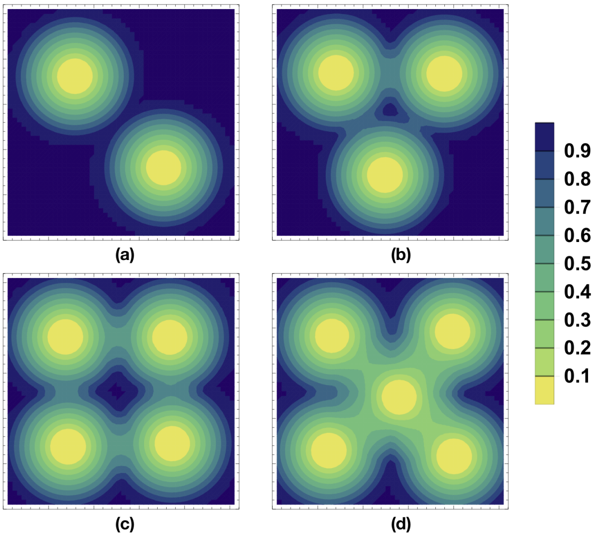

Fig. 5. The figure shows the search value over the whole surveillance area for different number of agents. As we can observe the agents take positions which result in the maximum area coverage (the figure shows the last time-step of a 50 time-step simulation experiment).

8.2 Results

The first set of experiments is conducted in order to investigate the proposed system on searching. More specifically the desired behavior is for the agents to jointly find the best configuration i.e. select the appropriate controls such that the total search value of the surveillance area is minimized or the area coverage is maximized. Our experimental setup is as follows. For a given number of agents we have conducted 50 Monte-Carlo (MC) trials where the agents are spawned at random locations (uniformly distributed) inside the surveillance area. Then we let our system to run for 50 time-steps and we monitor the total search value given by Eqn. (34). Our intuition is that the total search value should decrease over time since the agents would try to move to locations where the overlap of their sensing ranges, is minimized. In other words, we would like to have the agents spread as much as possible inside the surveillance region to increase coverage. Figure 4 shows the average total search value for each time-step of this experiment for 2, 3, 4, and 5 agents. As we can see from this figure, the total search value is decreasing over time, which indicates that the agents are trying to cover as much area as possible. It is also clear that as the number of deployed agents inside the surveillance region increases, the covered area increases as well.

In order to gain more insights into the behavior of the agents during searching, we have being monitoring the control actions assigned to them. At the end of the experiment, the final locations of the agents for the 4 cases are shown in Fig. 5. Interestingly, we can see that the agents take the appropriate formation which results in maximum coverage. For instance from Fig. 5a, we see that when the number of agents is 2 the optimal configuration would be to take the diagonal of the surveillance area. On the other hand when we have 3 and 4 agents, the optimal configuration forms a triangle and a square respectively, whereas in the case of 5 agents, the last agent moves into the middle of the area to fill the gap. The previous experiments showed that the solution to the problem of Eqn. (35) resulted in optimal coverage. In other words, depending on the number of agents, the obtained solution is a configuration of agent positions which maximizes the area coverage or minimizes the total search value.

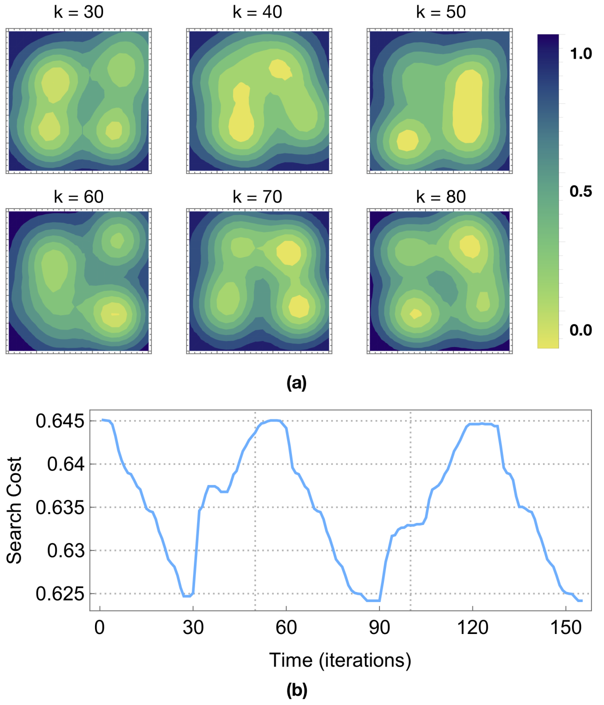

Fig. 6. Memory-based searching: (a) The figure shows the evolution of the search value over the whole surveillance area for various time-steps of a 180 time-step experiment. As we can observe the agents are continuously moving inside the surveillance area to cover unexplored areas. (b) the corresponding search cost over same experiment.

We should note here that once the agents obtain a configuration which provides the optimal coverage (Fig. 5),

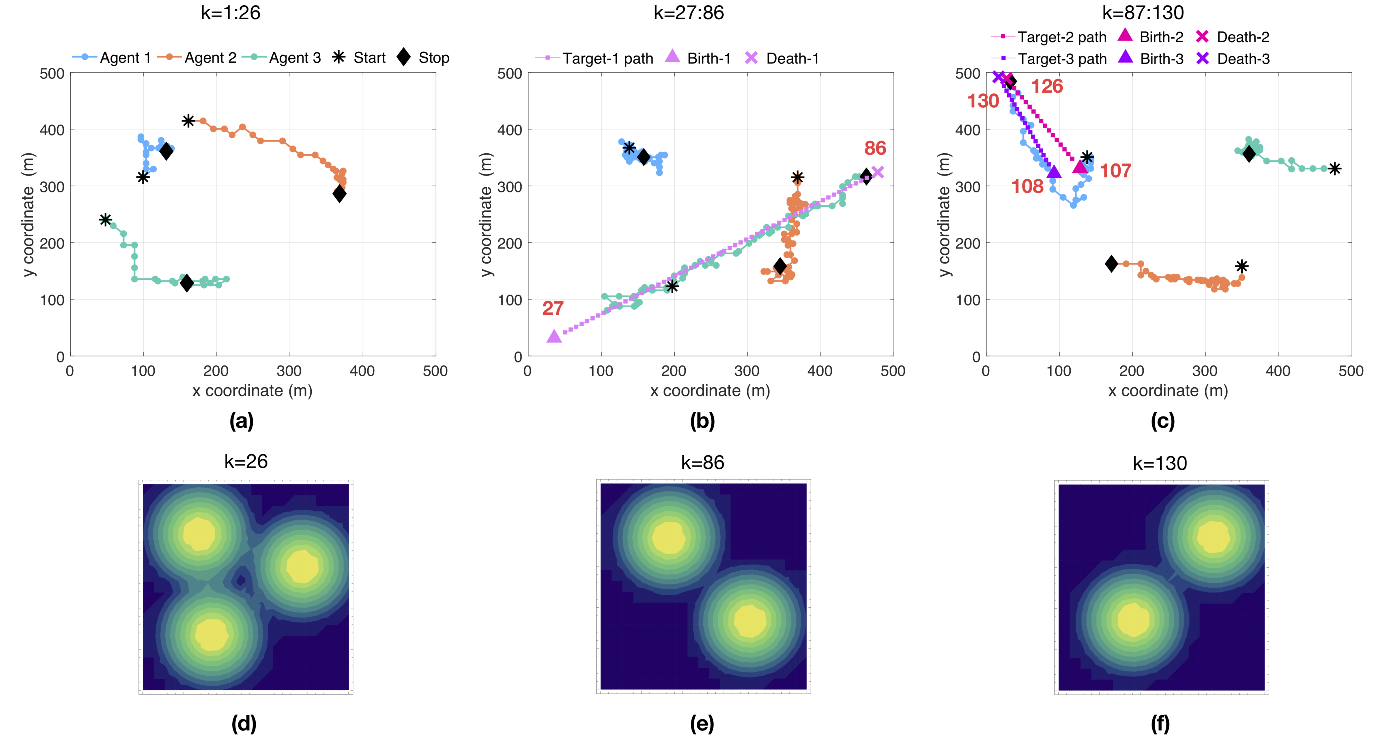

Fig. 7. (a) - (c): The figure shows the maneuvers of 3 agents for the task of joint search and track in a simulated scenario during 3 time periods. Agent start and stop positions are marked with $\star$ and $\Diamond$, respectively. Target birth and death positions are marked with $\triangle$ and $\times$ respectively and their corresponding birth/death times are shown in red. (d) - (f): The figure shows the search value over the surveillance area for 3 time-steps i.e. $k = 26, 86$ and 130. As we can observe the agents in *search* mode go to locations which result in the maximum area coverage.

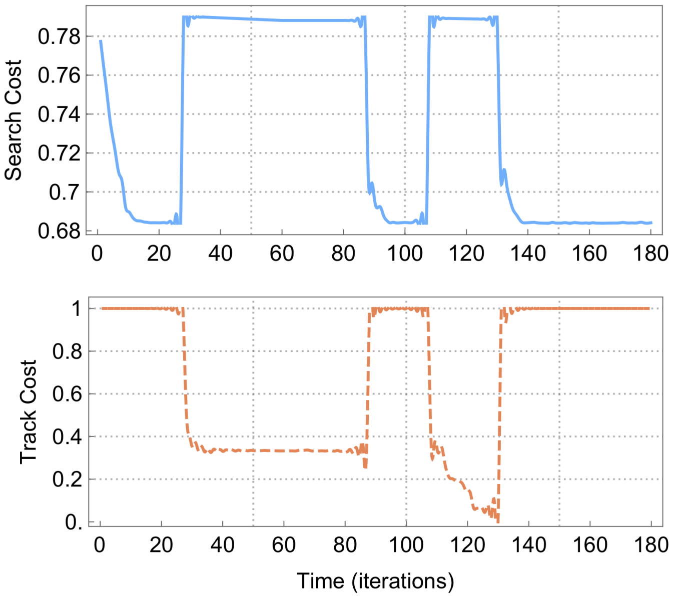

Fig. 8. The figure shows the values of the search and track objective functions for a simulated 180 time-step scenario. At each time-step, the proposed system decides based on the cost of search and track which mode to assign to each agent so that the coupled objective is minimized.

they lock in that configuration (i.e. there is no movement) until something is changed (e.g. a target appears). This is quite reasonable since the search function of Eqn. (39) does not provide incentive for any movements once the optimal coverage is obtained. However, in certain situations it would be desirable to have the agents move to areas that have not been visited for some time and temporarily take a configuration which does not provide the optimal coverage.

In order to achieve this we extend our proposed approach to include the notion of memory as follows: The total search value of Eqn. (39) now becomes the instantaneous total search value at time k. The new total search value is calculated as the weighted moving average of the κ previous instantaneous total search values where κ is the memory of our system. Figure 6 shows the memory-based searching behavior of our system for the case of 4 agents and $\kappa = 30$. More specifically, the figure shows the behavior of 4 agents in memory-based searching for 150 time-steps. In comparison with Fig. 5c we observe that the agents in this experiment are continuously moving in the surveillance area as shown in Fig. 6a. As before the 4 agents at time $k = 30$ take a square formation which gives the optimal coverage. However, since the system now takes into account the 30 previous time-steps, at time $k = 40, 50, ...$ we observe that a different formation has been taken. This is because the system *remembers* what areas have been visited and pushes the agents to new locations. At $k = 80$ the agents move again to form a square formation as shown in the figure. Fig. 6b shows the evolution of the total memory-based search cost for this experiment. As we can observe there is an evident periodicity in this graph which is due to the memory effect.

Next, in order to verify the effectiveness of the proposed approach on the task of joint searching and tracking, we have conducted several experiments with different numbers of targets and agents. In this section, we will give a representative example with 3 agents and 3 targets. This scenario is depicted in Fig. 7 during 3 different time-windows $k = 1..26$, $k = 27..86$ and $k = 87..130$. More specifically, 3 agents enter the surveillance area of size 500m × 500m at the locations marked with asterisks in Fig. 7a with coordinates $[100, 315]$,$[160, 415]$ and $[48, 240]$ for agents 1, 2 and 3, respectively. This experiment lasts 180 time-steps during which we have 3 target births and 3 target deaths. The ground-truth target birth/death times are $k = 27/86$, $k = 107/126$ and $k = 108/30$ for targets 1, 2 and 3, respectively. The target birth locations are $[35, 31]$, $[128, 331]$ and $[92, 321]$ for targets 1, 2 and 3 respectively and their corresponding death locations are $[478, 324]$, $[27, 489]$ and $[16, 492]$. In Fig. 7 the target birth locations are indicated by triangles, the target death locations are indicated by crosses and their birth/death times are shown in red. The trajectories of the agents and the targets during this experiment are shown in Fig. 7a - 7c for periods $k = 1..26$, $k = 27..86$ and $k = 87..130$ respectively.

The agents enter the surveillance area in search mode at $k = 1$ and at each time-step the proposed controller decides which mode (i.e. *search* or *track*) to assign to each agent. As we have already discussed each agent maintains a probability distribution regarding the number of targets and their location. The controller monitors the cost of tracking and the cost of searching and decides what are the optimal controls for all agents which minimize the objective function of Eqn. (35). To summarize the cost of tracking takes into account whether or not targets exist in the area, the uncertainty of the estimated number of targets in the area and finally how many and how well targets are being tracked by each agent. On the other hand, the cost of searching takes into account how well the area is covered by the agents.

Figure 7a shows the trajectories of all agents during period $k = 1..26$. During this period no targets exist in the area and so the controller decides to assign all agents in search mode. The objective now is to increase coverage and so the agents move away from each other, trying to find the locations where the coverage is maximized. Figure 7d shows the search value for each location in the surveillance area for $k = 26$, i.e. at the end of the period. As we can observe the agents take a triangle formation at the right locations which results in the maximum coverage. To understand better what is happening inside the controller, we have recorded the raw values of the search and track objective functions of Eqn. (39) and Eqn. (40) respectively for the whole experiment. This is shown in Fig. 8. The first point to note here is that the search cost is decreasing during time-steps $k = 1..26$, which indicates that the coverage is increasing. On the other hand, we can observe that the tracking cost during the same period is maximum, since the number of estimated targets in the area is 0. Thus the objective function of Eqn. (35) is minimized when all agents are in search mode.

Now, at time $k = 27$ target 1 appears in the area as shown in Fig. 7b which is then being detected by agent 3 at the next time-step $k = 28$. During time-steps $k = 28..35$ agent 3 moves towards target 1 as shown in the figure and then for the rest of the period $k = 36..86$, the same agent tracks target 1. It is also worth noting that during this whole period, agents 1 and 2 are in search mode, and are moving to locations which will result in optimal coverage. Figure 7e shows the search configuration of the two agents in *search* mode and the resulting search values for the whole area. Figure 8 sheds some more light into this process. During time-steps $k = 28..35$ we observe that the tracking cost is decreasing and afterwards during $k = 36..86$ to be reaching a plateau. The tracking cost is decreasing during $k = 36..86$ (i.e., the tracking accuracy is increasing) since the agent comes closer and closer to the target which allows the agent to perform better tracking. This is because the sensing and the measurement model depend on the distance of the agent to the targets. For distant targets, the uncertainty is higher thus the tracking accuracy lower. Once the target is locked in, the cost of tracking reaches a plateau. On the other hand, the search cost increases instantly since only 2 agents (out of 3) are left in search mode, and thus the coverage has been decreased. There is however a small improvement in the search cost during this period as the agents take a formation which results in better coverage.

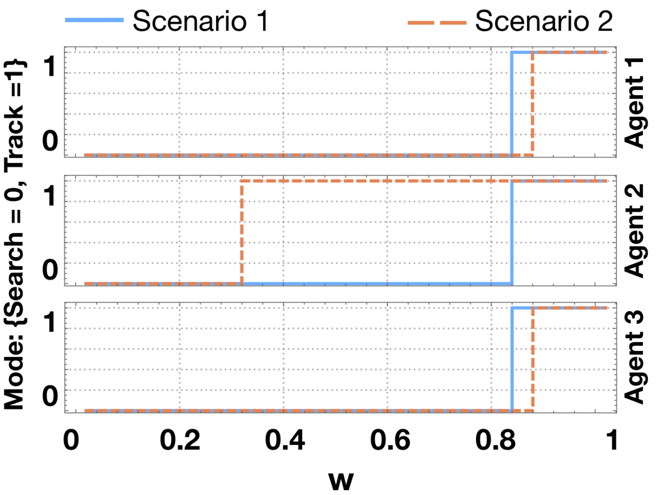

Fig. 9. The figure illustrates the effect of the parameter w on the search and track behavior of the system.

Target 1 dies at time-step $k = 86$, which drops the number of estimated targets, of agent 3, from 1 to 0 at $k = 87$. This increases the tracking cost to the maximum since no targets exist inside the surveillance area and as a consequence the controller assigns all agents to *search* mode. This is shown in Fig. 8 for $k = 87..107$. At $k = 107$ the agents take a formation similar to Fig. 7d. During period $k = 107..130$ two targets appear as shown in Fig. 7c which are being detected and tracked by agent 1. The tracking cost in Fig. 8 for this period reaches a new minimum since now two targets are being tracked and so the value of tracking over searching is increased. Agents 2 and 3 detect no target and remain in search mode moving towards locations which result in maximum coverage as shown in Fig. 7f. Finally, during period $k = 131..180$, no targets are being detected, thus the controller assigns all agents to search mode. The tracking cost reaches the maximum and the agents optimize the coverage as shown in Fig. 8.

Furthermore, we investigate hereafter, the impact of the parameter w of Eqn. (35) on the search-and-track behavior of the system. We assume the presence of 3 agents in locations $[100, 200]$, $[250, 150]$ and $[300, 400]$ for agents 1, 2 and 3, respectively, and we vary w between $[0...1]$ observing how this affects the operating mode (i.e. search or track) of each agent. We investigate 2 scenarios i.e. in the first scenario a target is located in position $[350, 50]$ and in the second scenario a target is located in position $[260, 140]$. As we can

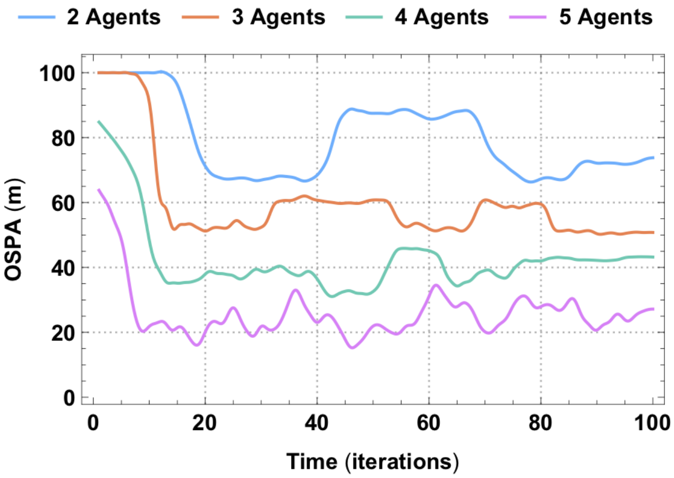

Fig. 10. The figure shows the performance of the proposed system for searching and tracking 10 targets with 2, 3, 4 and 5 agents.

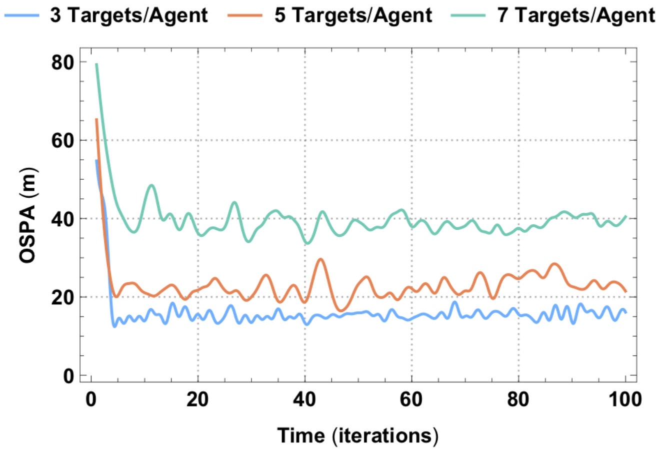

Fig. 11. The figure shows the OSPA error when 2 agents track 3, 5 and 7 targets each.

observe from Fig. 9, values of w that are close to 0 force all agents to switch to search mode whereas values of w very close to 1 have the opposite effect i.e. all agents are forced into track mode. This is a direct consequence of Eqn. (35). Interestingly, in Scenario 1 the target located in $[350, 50]$ is far away from all the agents and thus for moderate values of w close to 0.5 the system chooses to have all agents in search mode. In particular, in this scenario the target is detected by agent 2 however, the normalized tracking variance $\tilde{\sigma}$ of this agent is as high as 0.9 which results in very high tracking cost. This makes the system to prefer to have the particular agent in searching mode rather than in tracking mode for $w \leq 0.83$. In scenario 2 however, agent 2 is tracking the target located at $[260, 140]$ very accurately with $\tilde{\sigma} = 0.05$. As a consequence the system assigns agent 2 in tracking mode for $w \geq 0.3$ as is shown in Fig. 9. In this scenario however we can still force all agents in searching or tracking mode with the appropriate values for w. Dynamically adjusting w depending on a given situation is something interesting which we will investigate in a future work.

Moreover, we investigate the performance of the proposed approach with respect to the number of agents and the number of targets. In order to quantify the performance of the proposed approach we use the optimal sub-pattern assignment (OSPA) metric [48] which is widely used to measure the accuracy of multi-object filters. Our first experiment is conducted as follows: We randomly generate 10 targets inside the surveillance area and we use OSPA to quantify the effectiveness of the proposed search-and-track approach with 2, 3, 4 and 5 agents. All agents are spawned from the center of the surveillance area at the start of each trial. This is done in order to simultaneously test the performance of search (i.e. coverage) and track. Intuitively, we would like to see the agents to start acquiring a configuration which results in maximum coverage and while doing so increasing their probability of finding and tracking the scattered targets in the surveillance region. Figure 10 shows the average OSPA error (with parameters $c = 100$ and $p = 2$) over 30 Monte Carlo trials for 100 time-steps for different number of agents. As we can see in all scenarios the tracking error starts high but drops significantly as time progresses. The agents start from the center of the surveillance area and jointly perform searching by covering as much surveillance region as possible. This is evident by the high OSPA error in the beginning of the experiment. In particular, we observe in the case of 2 agents that no targets are detected between time-steps 0 and 17. During that time the agents maximize coverage (i.e. see Fig. 5a). Around time-step 16 the OSPA error begins to drop which indicates that targets are being tracked. Between time-steps 20 to 40 the 2-agent system reaches its tracking capacity and achieves its best performance. Between time-steps 40-70 we can see that the error has been increased. We have observed that when this happens the agents are tracking single targets due to targets splits and target deaths which increases the tracking error (while elsewhere other targets might have been born which have not been detected by the already occupied agents). However, when no targets are being tracked, or the targets tracked are far away, the agents switch to searching in order to optimize coverage. This results in a decrease in the tracking error between time-steps 70 to 100. Similar behavior is also true for 3, 4 and 5 agents. As the number of agents increases the time to detect targets decreases as is shown in the figure. Finally, we observe that the tracking capacity of the system increases with the number of the agents, as a consequence the tracking error decreases as the number of agents increases.

In the next experiment we investigate how the number of targets affect the performance of our system. More specifically, we have fixed the number of agents to 2 and we varied the number of targets tracked by each agent. Figure 11 shows the OSPA error when 2 agents track 3, 5 and 7 targets each for 100 time-steps. We should note here that in this experiment, the targets per agent (i.e. 3, 5 and 7) are initialized inside the corresponding agent's sensing range. Additionally the targets per agent move towards the same destination and without target splits occurring for the duration of the experiment. This is done here in order to investigate the ability of an agent to track 3, 5 and 7 targets. The figure shows the average error over 30 Monte Carlo trials. As we can observe, the tracking accuracy drops as the number of targets per agent increases. However, this experiment shows that the proposed approach is able to keep track multiple targets per agent. In particular, 3 and 5 targets per agent can be tracked with relatively high

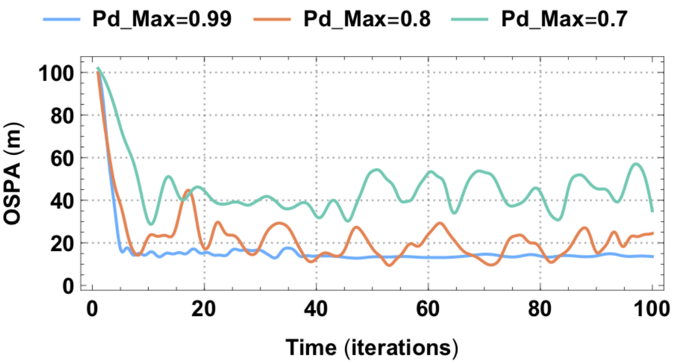

Fig. 12. The figure shows the OSPA error when 3 agents track 3 targets with varying $p_D^{\max}$ values.

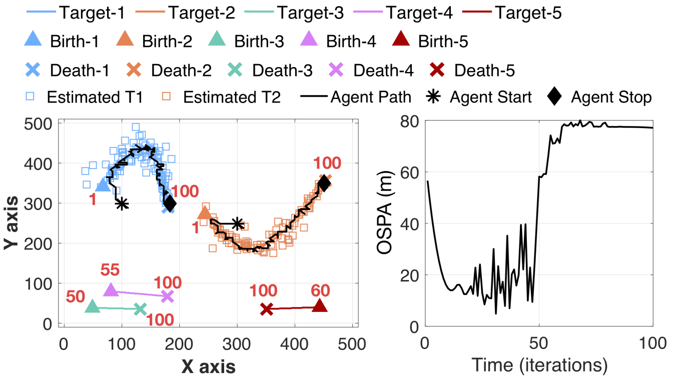

Fig. 13. The figure shows a search-and-track scenario which illustrates the limitations of the proposed approach.

accuracy in this scenario.

Up to this point, all of our experiments were based on the sensing model of Eqn. (3). In this model the parameter $p_D^{\max}$ determines the maximum detection probability of an agent with its value set to 0.99. In order to investigate the impact of this parameter to the tracking accuracy of our system, 30 Monte-Carlo trials have been conducted of length 100 iterations, for 3 agents and 3 targets with varying values of $p_D^{\max}$. More specifically, the experimental set-up is as follows: for each Monte-Carlo run at time-step $k = 0$, three agents are randomly generated inside the surveillance area. Then for each agent at $k = 0$, a single target is spawned inside its sensing range. We leave the system to run for 100 time-steps and we measure the OSPA error. Figure 12 shows the average OSPA error over 30 trials for $p_D^{\max} = 0.99$, $p_D^{\max} = 0.8$ and $p_D^{\max} = 0.7$. As we can observe the best accuracy is achieved with a $p_D^{\max}$ value of 0.99. On the other hand for $p_D^{\max} = 0.8$ and $p_D^{\max} = 0.7$ we observe significant OSPA oscillations which occur due to target miss-detections. More specifically, we have observed that a target miss-detection especially in the area beyond the agent's primary radius R_0 (where the probability of detection drops below $p_D^{\max}$) results in a) increased positioning error and b) the agent to take incorrect control action and/or switch mode (from tracking to searching) in subsequent time-steps which causes cardinality errors. Finally, note that the multi-Bernoulli filter (i.e. Eqn. (19)-(25)) is best suited for situations where the signal to noise ratio is high i.e. high probability of detection and the clutter rate is low. These problems can be alleviated with the more accurate labeled multi-Bernoulli (LMB) filter [38]. The LMB filter however, is computationally more expensive.

Finally, we conclude the evaluation of the proposed approach by investigating some of its limitations. For this purpose we use the following setup: We generate 5 targets with initial positions $[67, 341]$, $[244, 272]$, $[49, 38]$, $[81, 79]$ and $[443, 40]$ for targets 1, 2, 3, 4 and 5, respectively. The birth/death times for the 5 targets are $k = 1/100$, $k = 1/100$, $k = 50/100$, $k = 55/100$ and $k = 60/100$. Additionally at $k = 0$ the position of two agents are $[100, 298]$ and $[300, 248]$ for agents 1 and 2 respectively. This set-up is shown in Fig. 13. At $k = 1$ both agents detect their nearby targets (target 1 and 2) and switch to track mode. The figure shows with blue and orange squares the estimated target positions. Initially the estimated target states exhibit large positioning errors however, as the figure shows, in subsequent time-steps the estimation error improves as the agents have locked-in the targets. The agent trajectories are shown with black lines. We should note here that in this scenario we use the linear target dynamics described in the beginning of this section to track the non-linear motion of target 1 and target 2. It is evident from Fig. 13 that during time-steps 20 to 50 the OSPA error exhibits high oscillations which is due to the non-linear target behavior during that time-window. This is also shown by the high variance in the estimated target positions (blue and orange squares). At time-steps 50, 55 and 60 targets 3, 4 and 5 appear in the surveillance region following the trajectories shown in the figure. These targets, however are not detected by any of the agents since both agents are occupied tracking target 1 and 2. As we can observe the OSPA error increases dramatically from $k = 50$ onwards. This example shows the main limitation of the proposed system i.e. agents remain in a particular mode indefinitely. One way to approach this problem is by dynamically controlling the w parameter and periodically forcing an agent to enter either the search or track mode depending on the situation. Another way will be to keep track of the visited areas in the surveillance region and store statistics regarding the time of visit, the number of targets detected at each area, etc. and then perform a priority-based search in order to maximize the number of detected targets. These ideas will be explored in future works.

9 Conclusion and Future Work

In this paper we have studied the problem of jointly optimized searching and tracking with multiple agents in stochastic environments. We have presented a novel unified probabilistic framework for this decision and control problem based on Bayesian multi-object stochastic filtering with random finite sets. We have defined suitable objective functions for the tasks of multi-agent searching and tracking and we have showed that the resulting non-linear binary program can be approximated adequately by a genetic algorithm. Finally, we have demonstrated the performance of the proposed decision and control algorithm through extensive numerical experimentation. Future work will look at simpler and computationally improved approximations of the non-linear binary program as well as alternative and simpler objective functions with equivalent behavior and performance. In addition, we are interested in investigating a rolling horizon (i.e. multiple-step look ahead) predictive control approach for this problem and analyzing its performance against the proposed single-step look ahead system. We are also interested in investigating this problem with heterogeneous agents i.e. agents which exhibit different types of sensing models.

ACKNOWLEDGMENTS

This work is supported by the European Union Civil Protection under grant agreement No 783299 (SWIFTERS), by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

REFERENCES

[1] R. P. S. Mahler, "Statistics 101 for Multisensor, Multitarget Data Fusion," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 53–64, Jan 2004.

[2] R. Mahler, "Statistics 102 for Multisource-Multitarget Detection and Tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 376–389, June 2013.

[3] V. Mersheeva and G. Friedrich, "Multi-UAV Monitoring with Priorities and Limited Energy Resources," in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[4] A. Khan, E. Yanmaz, and B. Rinner, "Information Exchange and Decision Making in Micro Aerial Vehicle Networks for Cooperative Search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, Dec 2015.

[5] G. Avellar, G. Pereira, L. Pimenta, and P. Iscold, "Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time," *Sensors*, vol. 15, no. 11, p. 27783–27803, Nov 2015.

[6] H. Ding and D. Castañón, "Multi-agent Discrete Search with Limited Visibility," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 108–113.

[7] X. Sun, C. G. Cassandras, and X. Meng, "A Submodularity-based Approach for Multi-Agent Optimal Coverage Problems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 4082–4087.

[8] S. Rahili, J. Lu, W. Ren, and U. M. Al-Saggaf, "Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory: Algorithms and Experiments," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1303–1313, June 2018.

[9] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On Multi-Modal People Tracking from Mobile Platforms in very Crowded and Dynamic Environments," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5512–5519.

[10] S. Papaioannou, H. Wen, Z. Xiao, A. Markham, and N. Trigoni, "Accurate Positioning via Cross-Modality Training," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 239–251. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809712

[11] S. Papaioannou, A. Markham, and N. Trigoni, "Tracking People in Highly Dynamic Industrial Environments," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2351–2365, Aug 2017.

[12] R. Dutta, L. Sun, and D. Pack, "A Decentralized Formation and Network Connectivity Tracking Controller for Multiple Unmanned Systems," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–8, 2017.

[13] C. Xi and U. A. Khan, "Distributed Dynamic Optimization over Directed Graphs," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 245–250.

[14] F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch, "Distributed Estimation With Information-Seeking Control in Agent Networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2439–2456, Nov 2015.

[15] Y. Liu and G. Nejat, "Multirobot Cooperative Learning for Semiautonomous Control in Urban Search and Rescue Applications," *Journal of Field Robotics*, vol. 33, no. 4, pp. 512–536, 2016.

[16] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A Multirobot Path-Planning Strategy for Autonomous Wilderness Search and Rescue," *IEEE Trans. Cybernetics*, vol. 45, no. 9, pp. 1784–1797, 2015.

[17] Z. Beck, L. Teacy, A. Rogers, and N. R. Jennings, "Online Planning for Collaborative Search and Rescue by Heterogeneous Robot Teams," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 1024–1033.

[18] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An Autonomous multi-UAV System for Search and Rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, 2015, pp. 33–38.

[19] Y. Liu and G. Nejat, "Robotic Urban Search and Rescue: A Survey from the Control Perspective," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 2, pp. 147–165, Nov 2013.

[20] B. Shah and H. Choset, "Survey on Urban Search and Rescue Robots," *Journal of the Robotics Society of Japan*, vol. 22, no. 5, pp. 582–586, 2004.

[21] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, *Optimal Search for a Lost Target in a Bayesian World*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 209–222.

[22] C. Liu and J. K. Hedrick, "Model Predictive Control-based Target Search and Tracking using Autonomous Mobile Robot with Limited Sensing Domain," in *2017 American Control Conference (ACC)*, May 2017, pp. 2937–2942.

[23] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Autonomous Ground Target Tracking by Multiple Cooperative UAVs," in *2005 IEEE Aerospace Conference*, March 2005, pp. 1–9.

[24] G. M. Hoffmann, S. L. Wasl, and C. J. Tomlin, "Distributed Cooperative Search using Information-Theoretic Costs for Particle Filters with Quadrotor Applications," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2006, pp. 21–24.

[25] G. M. Hoffmann and C. J. Tomlin, "Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, Jan 2010.

[26] R. R. Pitre, X. R. Li, and R. Delbalzo, "UAV Route Planning for Joint Search and Track Missions: An Information-Value Approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2551–2565, JULY 2012.

[27] C. K. Peterson, "Dynamic Grouping of Cooperating Vehicles using a Receding Horizon Controller for Ground Target Search and Track Missions," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2017, pp. 1855–1860.

[28] Mahler, Ronald P. S., *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, Inc., 2007.

[29] S. Blackman and R. Popoli, "Design and Analysis of Modern Tracking Systems (Artech House Radar Library)," *Artech house*, 1999.

[30] Mahler, Ronald P. S., *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, Inc., 2014.

[31] B.-n. Vo, M. Mallick, Y. Bar-shalom, S. Coraluppi, R. Osborne III, R. Mahler, and B.-t. Vo, "Multi-target Tracking," *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2015.

[32] R. P. S. Mahler, "Multitarget Bayes Filtering via First-order Multitarget Moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct 2003.

[33] R. Mahler, "PHD Filters of Higher Order in Target Number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, October 2007.

[34] B. T. Vo and B. N. Vo and A. Cantoni, "The Cardinality Balanced Multi-Target Multi-Bernoulli Filter and its Implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, Feb 2009.

[35] B. Vo and B. Vo, "A random Finite Set Conjugate Prior and Application to Multi-Target Tracking," in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Dec 2011, pp. 431–436.

[36] ——, "Labeled Random Finite Sets and Multi-Object Conjugate Priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, July 2013.

[37] B. Vo, B. Vo, and D. Phung, "Labeled Random Finite Sets and the Bayes Multi-Target Tracking Filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6554–6567, Dec 2014.

[38] S. Reuter, B. Vo, B. Vo, and K. Dietmayer, "The Labeled Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, June 2014.

[39] B. Vo, B. Vo, and H. G. Hoang, "An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1975–1987, April 2017.

[40] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[41] B. Vo, B. Vo, and A. Cantoni, "On Multi-Bernoulli Approximations to the Bayes Multi-Target Filter," in *Proc. Int'l Symp. Information Fusion*, 2007.

[42] B. Ristic, B. N. Vo, and D. Clark, "A Note on the Reward Function for PHD Filters with Sensor Control," *IEEE Transactions on*

Aerospace and Electronic Systems, vol. 47, no. 2, pp. 1521–1529, April 2011.

[43] M. Beard, B. T. Vo, B. N. Vo, and S. Arulampalam, "Sensor Control for Multi-Target Tracking using Cauchy-Schwarz Divergence," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 937–944.

[44] R. Mahler, "Multitarget Sensor Management of Dispersed Mobile Sensors," *Series on Computers and Operations Research, World Scientific (2004)*, 2004.

[45] E. Çela, *The Quadratic Assignment Problem*. Springer US, 1998.

[46] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[47] B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo Methods for Multitarget Filtering with Random Finite Sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, Oct 2005.

[48] D. Schuhmacher, B. T. Vo, and B. N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug 2008.

Savvas Papaioannou obtained his B.S. degree in Electronic and Computer Engineering from Technical University of Crete, his M.S degree in Electrical Engineering from Yale University and his Ph.D. degree in Computer Science from the University of Oxford. He is currently a Research Associate at the KIOS Research and Innovation Center of Excellence at the University of Cyprus. His research interests include intelligent systems, sensing platforms and architectures, sensor networks and sensor fusion, multi-target tracking and state estimation. He is a member of the IEEE and the ACM and also a reviewer for various journals.

Panayiotis Kolios is a Research Associate at the KIOS Research and Innovation Center of Excellence at the University of Cyprus. He received his B.Eng and Ph.D degrees in Telecommunications Engineering from King's College London in 2008 and 2011, respectively. His interests focus on both basic and applied research on networked intelligent systems. Some examples of such systems include intelligent transportation systems, autonomous unmanned aerial systems, and the plethora of cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency response aspects in which faults and attacks could cause disruptions that need to be effectively handled. He is an active member of IEEE, contributing to a number of technical and professional activities within the Association.

Theocharis Theocharides holds a Ph.D. degree in Computer Science and Engineering from the Pennsylvania State University (2006). He is currently an Associate Professor at the Department of Electrical and Computer Engineering, and the Director of Research at the KIOS CoE, both at the University of Cyprus. His research focuses on the broad area of intelligent embedded systems design, with emphasis on domain-specific architectures, evolvable and reconfigurable hardware, and design of reliable and low power embedded and application specific processors and circuits. He is a senior member of the IEEE and the IEEE Computer Society, an ACM Member, and a member of the HiPEAC Network of Excellence. Currently, he serves on the editorial boards of the IEEE Consumer Electronics Magazine, the ACM Journal on Emerging Technologies in Computing Systems, IEEE Design and Test magazine, and on several Organizational and Technical Program Committee boards of various IEEE/ACM Conferences.

Christos Panayiotou has received a B.Sc. and a Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst, in 1994 and 1999 respectively. He also received an MBA from the Isenberg School of Management, at the aforementioned university in 1999. Currently he is an Associate Professor at the Electrical and Computer Engineering Department at UCY and he serves as the Deputy Director of the KIOS CoE for which he is also a founding member. His research interests include distributed control systems, wireless, ad hoc and sensor networks, computer communication networks, quality of service (QoS) provisioning, optimization and control of discrete-event systems, resource allocation, simulation, transportation networks and manufacturing systems. He is a senior member of the IEEE and also a reviewer for various conferences and journals, and he has served in the organizing and program committees of various international conferences.

Marios M. Polycarpou is a Professor of Electrical and Computer Engineering and the Director of the KIOS Research and Innovation Center of Excellence at the University of Cyprus. He received the B.A degree in Computer Science and the B.Sc. in Electrical Engineering, both from Rice University, USA in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, in 1989 and 1992 respectively. His teaching and research interests are in intelligent systems and networks, adaptive and cooperative control systems, computational intelligence, fault diagnosis and distributed agents. Dr. Polycarpou has published more than 300 articles in refereed journals, edited books and refereed conference proceedings, and co-authored 7 books. He is also the holder of 6 patents. Prof. Polycarpou is a Fellow of IEEE and IFAC. He is the recipient of the 2016 IEEE Neural Networks Pioneer Award. He received with his co-authors the 2014 Best Paper Award for the journal Building and Environment (Elsevier). Prof. Polycarpou served as the President of the IEEE Computational Intelligence Society (2012-2013), and as the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems (2004-2010). He is currently the President of the European Control Association (EUCA). Prof. Polycarpou has participated in more than 60 research projects/grants, funded by several agencies and industry in Europe and the United States, including the prestigious European Research Council (ERC) Advanced Grant.