Distributed Multi-robot Tracking of Unknown Clustered Targets with Noisy Measurements

Jun Chen[1], Philip Dames[2], and Shinkyu Park[1]

[1] King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia,
{jun.chen.1, shinkyu.park}@kaust.edu.sa
[2] Temple University, Philadelphia 19122, PA, USA,
pdames@temple.edu

Abstract. Distributed multi-target tracking is a canonical task for multi-robot systems, encompassing applications from environmental monitoring to disaster response to surveillance. In many situations, the distribution of unknown objects in a search area is irregular, with objects are likely to distribute in clusters instead of evenly distributed. In this paper, we develop a novel distributed multi-robot multi-target tracking algorithm for effectively tracking clustered targets from noisy measurements. Our algorithm contains two major components. Firstly, both the instantaneous and cumulative target density are estimated, providing the best guess of current target states and long-term coarse distribution of clusters, respectively. Secondly, the power diagram is implemented in Lloyd's algorithm to optimize task space assignment for each robot to trade-off between tracking detected targets in clusters and searching for potential targets outside clusters. We demonstrate the efficacy of our proposed method and show that our method outperforms of other candidates in tracking accuracy through a set of simulations.

Keywords: Multiple target tracking, sensor-based control, distributed sensor network

1 Introduction

Multi-target tracking using distributed multi-robot systems (MRSs) has drawn increasing attention over the past decades as robots become more powerful and low-cost. In a large number of real-world scenarios, targets are likely to distribute in clusters despite lack of prior knowledge about exact locations, including social animals, certain species of plants, trash distributed in inhabited places, etc. In such cases, detecting a targets indicates that some other targets are likely to appear nearby. Most existing multi-robot multi-target tracking (MR-MTT) algorithms underperform in trading-off between having robots to search for un-detected targets and tracking detected targets when targets are not evenly distributed across the search space given no prior knowledge. This paper aims at developing effective distributed tracking algorithms for unknown targets that are likely to distributed in clusters. There are two key components to a multi-robot multi-target tracking (MR-MTT) system: an estimation system to model and track objects as they are detected and a control system to drive the motion of individual robots in the team towards areas that are likely to contain useful information.

Different from single target tracking, the main challenge of MTT is matching detections to target tracks, a process known as data association, especially in the presence of false negative and false positive detections. There are a number of standard MTT algorithms, each of which solve data association in a different way: global nearest neighbor (GNN) [9], joint probabilistic data association (JPDA) [8], multiple hypothesis tracking (MHT) [2], and particle filters [7]. Each of these trackers propagates the posterior of target states over time and solves the data association problem prior to tracking. Another class of MTT techniques is derived from random finite set (RFS) statistics [13] and which simultaneously solve data association and tracking. We use the probability hypothesis density (PHD) filter [12], which tracks the spatial density of targets, making it best suited to situations where each target is not required to have a unique identity. We recently developed a distributed PHD filter that is provably equivalent to the centralized solution [6].

Lloyd's algorithm [11] is one of the best-known control algorithms for distributed target tracking, the idea of which is to represent target states by a weighting function over the task space and to drive each robot to the weighted centroid of its Voronoi cell [5]. Comparing to other MR-MTT algorithms such as graph-based methods [1, 16], Lloyd's algorithm requires no prior knowledge about targets and the number of targets can be unknown and time-varying. It drives more robots to areas where targets are more likely to appear, while allows fewer robots to search for targets in the rest of the areas based on the weighting function. The weighting function can be estimated online using on-board sensors [15], and Dames in his work [6] uses the PHD as the weighting function, driving robots to actively track targets. However, when no target is within a robot's Voronoi cell, the robots move erratically, reacting to any false positive detections as well as the dynamically changing shape of their Voronoi cells. As a result, robots often stay within empty sub-regions instead of purposefully seeking out un-tracked targets, slowing down the rate at which they find targets. This problem is further exacerbated when a majority of targets gather within some small subsets of the environment.

To improve tracking accuracy of clustered targets, this paper develops a novel estimation and control policy. We have three primary contributions: 1) we introduce a state estimation strategy incorporating both instantaneous and cumulative target states, which allows robots to track detected targets precisely through noisy measurements while learning coarse distribution of targets from historical detecting outcomes, 2) we implement the power diagram in Lloyd's algorithm to dynamically assign optimized task space to each robot based on its current workload and drive the robots to explore or track targets more effectively, and 3) we demonstrate in a series of simulated experiments that with our proposed method, the team finds and tracks targets more effectively than using the previous algorithms in [6].

2 Problem Formulation

A set of nt targets with states $X = \{x_1, \ldots, x_{nt}\}$ are located within a convex open task space denoted by $E \subset R^2$. A team of nr (possibly heterogeneous) robots $R = \{r_1, \ldots, r_{nr}\}$ are tasked with determining nt and X, both of which are unknown and may vary over time. We assume that each robot r_i knows its location q_i in a global reference

frame (e.g. from GPS), though our proposed method can be immediately extended to handle localization uncertainty using the algorithms from our previous work [3]. At each time step, a robot r_i receives a set of $|Z_i|$ noisy measurements $Z_i = \{z_i, z_2, \ldots\}$ of targets within the field of view (FoV) F_i of its onboard sensor. Note that the sensor may experience false negative or false positive detections so the number of detections may not match the true number of targets.

2.1 Lloyd's Algorithm

The objective of Lloyd's algorithm is to minimize the following functional:

$$\mathcal{H}(Q, \mathcal{W}) = \sum_{i=1}^{n} \int_{\mathcal{W}_i} f\left(\|x - q_i\|\right) \phi(x) dx, \tag{1}$$

where $\mathcal{W}_i$ is dominance region of robot r_i (*i.e.*, the region that robot r_i is responsible for), $\|\cdot\|$ is the Euclidean distance, $x \in E$, and $f(\cdot)$ is be a monotonically increasing function. The role of f is to quantify the cost of sensing due to degradation of a robot's ability to measure events with increasing distance. The dominance regions $\mathcal{W}_i$ form a partition over E, meaning the regions have disjoint interiors and the union of all regions is E [5].

The goal is for the team to minimize the functional (1), both with respect to the partition set $\mathcal{W}$ and the robot positions Q. Minimizing $\mathcal{H}$ with respect to $\mathcal{W}$ induces a partition on the environment $V_i = \{x \mid i = \arg\min_{k=1,\ldots,n} \|x - q_k\|\}$. In other words, V_i is the collection of all points that are the nearest neighbor of r_i. This is the Voronoi partition, and these V_i are the Voronoi cells, which are convex by construction. We call q_i the generator point of V_i. Minimizing $\mathcal{H}$ with respect to Q leads each sensor to the weighted centroid of its Voronoi cell [5], that is

$$q_i^* = \frac{\int_{V_i} x\phi(x)\, dx}{\int_{V_i} \phi(x)\, dx}. \tag{2}$$

Lloyd's algorithm sets the control input for robot r_i to $u_r(q_i^*)$, where

$$u_i(g) = \min\left(d_{\text{step}}, \|g - q_i\|\right) \frac{g - q_i}{\|g - q_i\|}, \tag{3}$$

g is an arbitrary goal location, and $d_{\text{step}} > 0$ is the distance a robot can move during one time step. By following this control law, the sensors asymptotically converge to the weighted centroids of their Voronoi cells, causing the team to reach a local minimum of Equation (2). This still holds true when $\phi(x)$ varies with time.

3 Distributed Multi-target Tracking

3.1 Instantaneous State Estimation

The sets X and Z_i from above contain a random number of random elements, and thus are realization of random finite sets (RFSs) [13]. The first order moment of an RFS is

known as the *Probability Hypothesis Density* (PHD) (which we denote $v(x)$) and takes the form of a density function over the state space of a single target or measurement. By assuming that the RFSs are Poisson distributed where the number of targets follows a Poisson distribution and the spatial distribution of targets is i.i.d., the PHD filter recursively updates this target density function in order to track the distribution over target sets [12].

The PHD filter uses three models to describe the motion of targets: 1) The motion model, $f(x \mid \xi)$, describes the likelihood of an individual target transitioning from an initial state ξ to a new state x. 2) The survival probability model, $p_s(x)$, describes the likelihood that a target with state x will continue to exist from one time step to the next. 3) The birth PHD, $b(x)$, encodes both the number and locations of the new targets that may appear in the environment.

The PHD filter also uses three models to describe the ability of robots to detect targets: 1) The detection model, $p_d(x \mid q)$, gives the probability of a robot with state q successfully detecting a target with state x. Note that the probability of detection is identically zero for all x outside the sensor FoV. 2) The measurement model, $g(z \mid x,\ q)$, gives the likelihood of a robot with state q receiving a measurement z from a target with state x. 3) The false positive (or clutter) PHD, $c(z \mid q)$, describes both the number and locations of the clutter measurements.

Using these target and sensor models, the PHD filter prediction and update equations are:

$$\bar{v}_t(x) = b(x) + \int_E f(x \mid \xi) p_s(\xi) v_{t-1}(\xi)\, d\xi \tag{4}$$

$$v_t(x) = (1 - p_d(x \mid q))\bar{v}_t(x) + \sum_{z \in Z_t} \frac{\psi_{z,q}(x)\bar{v}_t(x)}{\eta_z(\bar{v}^t)} \tag{5}$$

$$\eta_z(v) = c(z \mid q) + \int_E \psi_{z,q}(x) v(x)\, dx \tag{6}$$

$$\psi_{z,q}(x) = g(z \mid x, q) p_d(x \mid q), \tag{7}$$

where $\psi_{z,q}(x)$ is the probability of a sensor at q receiving measurement z from a target with state x.

The PHD represents the best guess of target states at current time step and is utilized as the instant state estimation of target density, denoted by $wi(x)$, i.e., $wi(x) = v(x), x \in E$. In [6] a distributed PHD filter is formulated, with each robot maintaining the PHD within a unique subset, V_i, of the environment. Three algorithms then account for motion of the robots (to update the subsets V_i), motion of the targets (in (4)), and measurement updates (in (5)). In this paper, same strategies are applied for propagating $wi(x)$ over time in a distributed manner.

3.2 Cumulative State Estimation

Since no prior of birth model is known for the PHD filter, we typically assume that the target birth rate is uniformly distributed. As a result, individuals do not actively search for a target when no target is detected and could often spend a long time locating targets

that appeared in underexplored regions of the environment, degrading the tracking performance especially when targets are clustered. To improve this, we need to estimate online the probability of target appearance over a long period of time, i.e., the cumulative state wc.

In a variety of scenarios, targets move randomly in some relatively fixed clusters, e.g., animals cluster around water sources. In such cases, the frequency of target appearance at each point $x \in E$ can be regarded as time-invariant and the cumulative state estimation is a density distribution that quantifies the best guess of the number of expected targets at each location based on accumulated observation.

The PHD filter assumes that the number of targets follows a Poisson distribution. The conjugate prior of this is a Gamma distribution, which describes the distribution $f(x)$ of the expected number of targets found at x. This is given by

$$\Gamma(\alpha, \beta) = \frac{x^{\alpha-1} e^{-\beta x} \beta^{\alpha}}{(\alpha - 1)!},\ x > 0,\ \alpha, \beta > 0, \tag{8}$$

where $\alpha(x)$ is the shape parameter, which describes the total rewards obtained from the observations at location x, and $\beta(x)$ is the inverse rate parameter, which describes the inverse of the number of historical observations sensors conduct at location x. Initially, $\alpha_0(x) = 0$ and $\beta_0(x) = \infty$. When a robot r_i receives a measurement set Z_i at time $t > 0$, it updates $\alpha(x), \beta(x)$ for all $x \in F_i$ by

$$\begin{aligned} \alpha(x) &= \alpha(x) + rw, \\ \beta(x) &= (\beta(x)^{-1} + 1)^{-1}, \end{aligned} \tag{9}$$

where

$$rw = \begin{cases} 0 & |Z_i| = 0 \\ 1 & \text{else} \end{cases}. \tag{10}$$

Then, for each $x \in E$, the cumulative state estimation is given by the mode of the distribution $\Gamma(\alpha(x), \beta(x))$

$$wc(x) = \begin{cases} \frac{\alpha(x)-1}{\beta(x)}, & \alpha(x) \geq 1 \\ 0, & \alpha(x) < 1. \end{cases} \tag{11}$$

3.3 Environment Approximation

Since it is neither feasible or physically meaningful to have a different distribution for each point in space, it makes sense to use a subset of points to represent both wi and wc. In this paper, we use uniform grids to represent wi and wc over the task space. Therefore, the integral of wi and wc in a grid is equal to the instantaneous and cumulative expected number of targets in the grid, respectively.

3.4 Distributed Control

Optimized Space Assignment In order to optimize the assigned coverage area, each robot is endowed a weight ρ^2 in constructing it Voronoi cell, and the monotonically

increasing function in Equation (1) becomes

$$f(\|x - q_i\|) = \|x - q_i\|^2 - \rho_i^2 \tag{12}$$

where ρ_i is the power radius of robot r_i. The resulting $\mathcal{W}$ is a additively weighted Voronoi diagram, or power diagram, a variant of the standard Voronoi diagram, and $\mathcal{W}_i$ is the power cell. For each robot r_i, we set ρ_i as

$$\rho_i = \left(\int_{x \in \mathcal{W}_i} wc(x) \right)^{-\frac{1}{2}}, \tag{13}$$

so that the area of $\mathcal{W}_i$ is approximately inversely proportional to the estimated number of targets can be found in $\mathcal{W}_i$. Thereby, robots in areas with lower expected target density are assigned larger area to explore, which takes advantage of the search and tracking capability of the team.

3.5 Algorithm Outline

The distributed target search and tracking algorithm is outlined as Algorithm 1. For each robot, both instantaneous and cumulative states are initialized to null as there is no prior knowledge about target states, and $\alpha(x)$ and $\beta(x)$ are initialized. As robots start to explore and receive sensor measurements, each of them exchange its location with neighbors to compute its power cell $\mathcal{W}_i$, as outlined in lines 5-8. Then each robot updates and broadcast the tuple $\{\alpha(x), \beta(x), wi_i(x), wc_i(x)\}$. Lines 12-18 outline the control strategy, which separates the team into idle robots (i.e., robots that find no target) and busy robots (i.e., robots that find targets), and causes the robot to move according to the following rules: When targets are detected, it drives the robot to locations with higher instantaneous estimation of target density to better track the detected targets; On the contrary, a robot is driven to search for targets in areas with higher cumulative estimation of target density where targets are more likely to be found based on accumulated experience. As a result, the team is able to optimize robot locations by taking advantage of historical measurements and further improve the tracking accuracy of targets.

Algorithm 1 boosts the performance of our previous methods [6] in that it allows the team to actively explore the environment and learn the characteristics of the target distribution. In particular, robots are now able to use a combination of detailed local information (coming from the PHD) and coarse global information (coming from α, β) to inform their actions. For evenly distributed targets, Algorithm 1 yields a similar tracking performance compared with Dames' method [6] since wc is close to uniform across the task space and $\mathcal{W}$ is similar to a standard Voronoi diagram since then. However, the advantages of our method is pronounced when targets are not uniformly distributed in the space but are instead grouped together within small regions. Under these circumstances, idle robots are especially helpful in learning the difference in target density among sub-regions and optimizing the assignment of tracking effort in different sub-regions.

Algorithm 1: Distributed Search and Tracking

for $r_i \in R$ **do**
  $wa(x) \leftarrow 0, wi(x) \leftarrow 0, x \in E$
  $\alpha(x) \leftarrow \frac{1}{\infty}, \beta(x) \leftarrow \infty, x \in E$
while true **do**
  for $r_i \in R$ **do**
    Receive measurement set Z_i
    Exchange state q_i with neighbors $\mathcal{N}_i$
    Compute power cell $\mathcal{W}_i$
    Update $\alpha(x), \beta(x), x \in \mathcal{W}_i$ using (9)
    Update $wi_i(x), wc_i(x), x \in \mathcal{W}_i$ using (5) and (11), respectively
    Broadcast $\{\alpha(x), \beta(x), wi_i(x), wc_i(x)\}$
    if $Z_i = \emptyset$ **then**
      $\phi_i(x) \leftarrow wc_i(x), x \in \mathcal{W}_i$ ▷ Idle robots
    else
      $\phi_i(x) \leftarrow wi_i(x), x \in \mathcal{W}_i$ ▷ Busy robots
    Set goal $g_i = q_i^*$ using (2)
    Compute $u_i(g_i)$ using (3)
    $q_i \leftarrow q_i + u_i$ ▷ Move towards goal

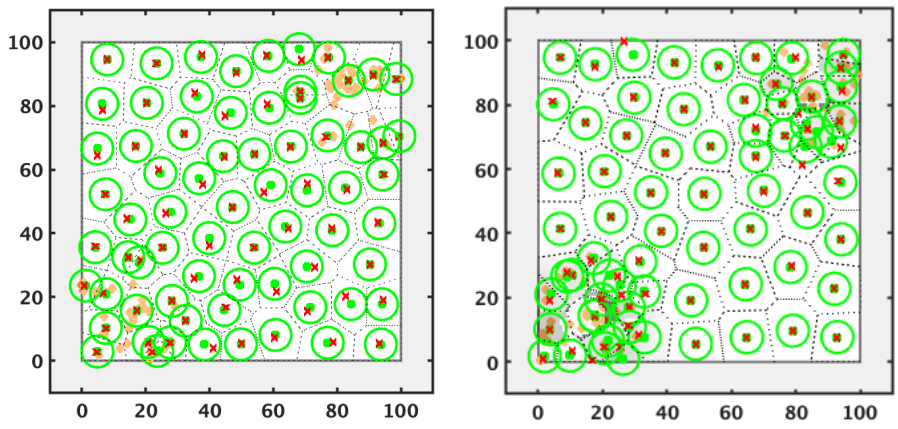
4 Simulations

We test our proposed algorithms via MATLAB simulations. The task space is a 100 m × 100 m square. Targets are distributed in clusters, where 30 are located in a 33 × 33 m square sub-region at the lower-left corner of E, and another 30 targets in a 33 × 33 m squared sub-region at the top-right corner. Targets move within the sub-regions following a Gaussian random walk at a maximum speed of 3 m/s. All robots begin each trial at randomized locations within a 20 m × 10 m box at the bottom center of the environment. Robots have a maximum speed of 5 m/s and are equipped with isotropic sensors with a sensing radius 5 m.

Both instantaneous and cumulative estimation are approximated by uniform grid implementation, with a grid size of 0.1 m × 0.1 m. In the PHD filter, the robots use a Gaussian random walk (with $\sigma = 0.35$ m/s) for the motion models f, set the survival probability to 1, and the birth PHD to 0 due to no prior knowledge of targets is learned. We use the same measurement model for homogeneous noisy sensors as [6]. Note that our proposed method is compatible with heterogeneous sensing network [4], we just make this choice to simplify the tests.

4.1 Qualitative Comparison

We first show how our proposed algorithms improve multi-target tracking using a single trial using 70 robots. Figures 1 show the locations of robots and targets after 300 s using both Dames' method [6] and our method. When targets gather within only a small portion

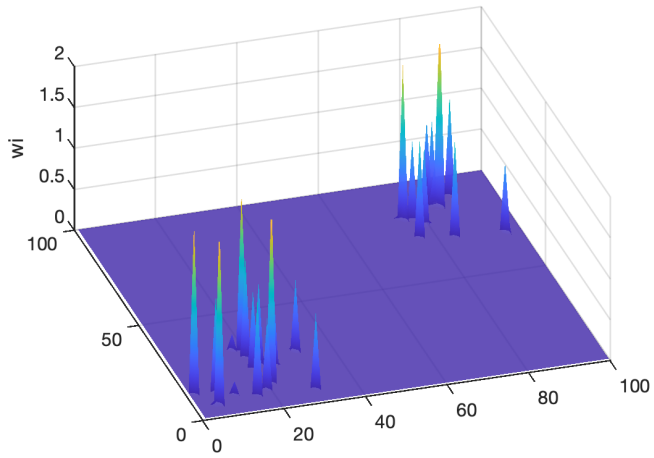


(a) Dames' Method (b) Our Method

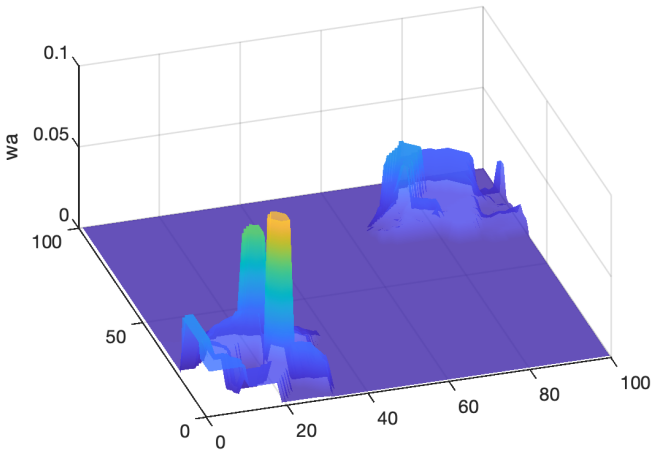
Fig. 1. Figures show 70 robots and 60 targets distribution in a 100 m × 100 m squared task space after 300 s of tracking using Dame's method and our method, respectively. Orange diamonds plot locations of targets. Green squares and circles show locations and field of views of robots, respectively. Red crosses plot robots' temporary goals. Dashed lines plot boundaries of robots' current assigned space.

of the environment, the previous method can hardly optimize robot locations for more accurate tracking of targets since only a few robots have found targets and are tracking them. As a result, most of the robots are idle and move erratically to cover areas where no targets are distribute, which demonstrates the weakness of the previous algorithm that idle robots do not actively search for targets, causing an inefficient use of the total sensing capability of the team while searching for un-tracked targets. Such phenomenon is improved by our method, which drives a larger number of robots to gather at the clusters of targets while a handful of other robots continue to search unexplored areas, as the exact locations and motion models of targets are unknown. The result indicates the efficacy of our method in that the distribution of the clusters is learned overtime and the power diagram distributes more balanced workload to individuals.

To illustrate the different characteristics of the instantaneous estimation and the cumulative estimation of targets, we plot in Figures 2 the value of $wi(x)$ and $wc(x)$ for all $x \in E$ after 300 s of the task using our method, respectively. The instantaneous estimation, i.e., the PHD, provides the best guess of exact locations of targets at current time through a set of sharp peaks, shown as Figure 2a. The PHD $wi(x)$ of $x \in E$ returns to near zero rapidly as targets are no longer found at x over a very short period of time given low expected target birth rate, exhibiting high accuracy of estimating target exact locations but short "memory" of historical target distribution. On the contrary, the cumulative estimation wc presents a relatively smooth and continuous distribution over the task space, with higher values distributed over the entire target clusters and near-zero values over the rest of the space, shown as Figure 2b. Therefore, the instantaneous and the cumulative estimation are utilized to drive busy and idle robots respectively, as the



(a) *wi*



(b) *wa*

Fig. 2. Figures plot surface of *wi* and *wc* distributed over the 100 m × 100 m task spaces after 300 s, respectively.

former robots requires target exact locations for accurate tracking while the latter robots are in need of coarse distribution of clusters for optimized deployment.

4.2 Quantitative Comparison

To test the efficacy of our proposed approach, we conduct a series of trials using a range of team sizes (from 60 to 90 robots). Three tracking strategies are compared: 1) Dames' method ("D" method) in [6], 2) Dames' method with power diagram ("P" method) which uses instantaneous estimation only for tracking (similar to the power diagram implementation in [4]), and the power radius in Equation (14) is depending on PHD instead, given by

$$\rho_i = \left(\sum_{x \in \mathcal{W}_i} wi(x) \right)^{-\frac{1}{2}}, \tag{14}$$

and 3) our method ("O" method) which uses both instantaneous and cumulative estimation, outlined in Algorithm 1. For each team size we run 10 trials.

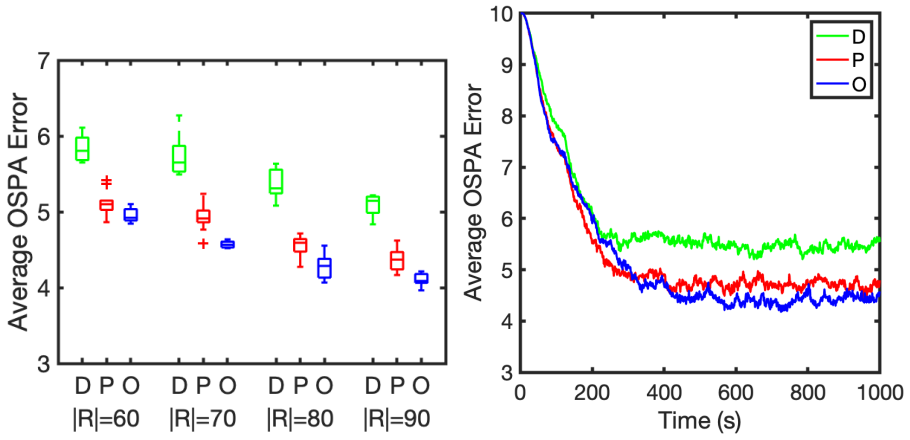
OSPA To quantify the performance, we will use the first order Optimal SubPattern Assignment (OSPA) metric [14], a commonly-used approach in MTT. The error between two sets X, Y, where $|X| = m \leq |Y| = n$ without loss of generality, is

$$d(X,Y) = \left(\frac{1}{n} \min_{\pi \in \Pi_n} \left(\sum_{i=1}^{m} d_c(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{1/p}, \tag{15}$$

where c is a cutoff distance, $d_c(x,y) = \min(c, \|x-y\|)$, and Π_n is the set of all permutations of the set $\{1, 2, \ldots, n\}$. This gives the average error in matched targets, where OSPA considers all possible assignments between elements $x \in X$ and $y \in Y$ that are within distance c of each other. This can be efficiently computed in polynomial time using the Hungarian algorithm [10]. We use $c = 10\,\text{m}$, $p = 1$, and measure the error between the true and estimated target sets. Note that a lower OSPA value indicates a more accurate tracking of the target set.

Results In Figure 3a, we report the median OSPA value to measure tracking accuracy over the final 700 s of each trial, allowing the team to reach or be close to a steady state and smoothing out the effects of spurious measurements that cause the OSPA to fluctuate, and the results are aggregated into boxplots. "D" method shows the worst performance in tracking clustered targets regardless of team sizes. As depicted by the results of "P" method, such performance is improved by assigning optimized task spaces that contain equivalent instantaneous estimated number of targets to robots in the team, which drives the robots to gather at where targets are currently tracked. However, robots are still not tend to move to where targets are likely to clustered by no target is tracked instantly. Our proposed method further improves this flaw and shows the best tracking performance of clustered targets, as suggested by the OSPA values of "O" method.

In Figure 3b, we plot the average OSPA values of all 40 trials of four different team sizes using the three methods over the entire 1000 s. It is shown that the OSPA error



(a) Steady State OSPA (b) OSPA VS Time

Fig. 3. Figures show OSPA errors of group of trials. Figure 3a displays boxplots of average OSPA errors from 300 s to 1000 s using the three methods ("D", "P", and "O") with four team sizes, each over ten trials. $|R|$ denotes the number of robots. Figure 3b plots average OSPA errors of the total of 40 trials (10 for each team size) using the three methods over the entire 1000 s.

drops at similar rates over the first 200 s as robots start moving from the starting area and explore the entire search space, despite the applied tracking algorithm. After that, "D" method no longer improves the tracking accuracy and the team reaches a steady state, while the other two algorithms continues to result in lower down of the OSPA error. "P" method reaches a steady state at around 400 s, after which the team has completed updating a steady-state PHD and propagating optimized cell assignment. Meanwhile, the OSPA error continuously drops by using "O" method until approximately 600 s. This is due to the fact that the cumulative estimation of target number continuously grows as more observations are gain, leading to an enhanced congregation of robots in target clusters. It is illustrated that our proposed method results in a continuous learning behavior of cluster distribution as robots explore and contributes to the outperformance of tracking accuracy over the other two algorithms.

5 Conclusions

The tracking accuracy of existing distributed MR-MTT algorithms significantly decays when targets are clustered instead of evenly distributed across the task space. In this paper, we propose a novel distributed multi-target tracking algorithm that allows a team of robots to effectively track clustered targets, despite given no prior knowledge of target states. Each robot estimates both instantaneous and cumulative target density and dynamically optimizes its space assignment using a power diagram implementation of Lloyd's algorithm. As a result, robots are able to track detected targets precisely while congregate at target clusters by learning the coarse cluster distribution from past

observations. Simulation results suggest that our algorithm is superior to other candidates in effective tracking of clustered targets.

References

1. Ahmad, A., Tipaldi, G.D., Lima, P., Burgard, W.: Cooperative robot localization and target tracking based on least squares minimization. In: 2013 IEEE International Conference on Robotics and Automation, pp. 5696–5701. IEEE (2013)
2. Blackman, S.S.: Multiple hypothesis tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine **19**(1), 5–18 (2004)
3. Chen, J., Dames, P.: Collision-free distributed multi-target tracking using teams of mobile robot with localization uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2020)
4. Chen, J., Dames, P.: Distributed multi-target tracking for heterogeneous mobile sensing networks with limited field of views. In: IEEE International Conference Robotics and Automation (2021). Under review
5. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Transactions on robotics and Automation **20**(2), 243–255 (2004)
6. Dames, P.M.: Distributed multi-target search and tracking using the phd filter. Autonomous robots **44**(3), 673–689 (2020)
7. Doucet, A., Vo, B.N., Andrieu, C., Davy, M.: Particle filtering for multi-target tracking and sensor management. In: Proceedings of the Fifth International Conference on Information Fusion, vol. 1, pp. 474–481. IEEE (2002)
8. Hamid Rezatofighi, S., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I.: Joint probabilistic data association revisited. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3047–3055 (2015)
9. Konstantinova, P., Udvarev, A., Semerdjiev, T.: A study of a target tracking algorithm using global nearest neighbor approach. In: Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03), pp. 290–295 (2003)
10. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
11. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
12. Mahler, R.P.: Multitarget bayes filtering via first-order multitarget moments. IEEE Transactions on Aerospace and Electronic systems **39**(4), 1152–1178 (2003)
13. Mahler, R.P.: Statistical multisource-multitarget information fusion, vol. 685. Artech House Norwood, MA (2007)
14. Schuhmacher, D., Vo, B.T., Vo, B.N.: A consistent metric for performance evaluation of multi-object filters. IEEE Transactions on Signal Processing **56**(8), 3447–3457 (2008)
15. Schwager, M., Rus, D., Slotine, J.J.: Decentralized, adaptive coverage control for networked robots. The International Journal of Robotics Research **28**(3), 357–375 (2009)
16. Sung, Y., Budhiraja, A.K., Williams, R.K., Tokekar, P.: Distributed assignment with limited communication for multi-robot multi-target tracking. Autonomous Robots **44**(1), 57–73 (2020)