Active Multi-Target Search Using Distributed Thompson Sampling

Jun Chen and Philip Dames

Abstract—Distributed search and track is a canonical task for multi-robot system, encompassing applications from environmental monitoring to disaster response to surveillance. In many situations, the distribution of objects in a search area is irregular, with some areas having high object densities while other areas have low densities. In this paper, we propose a novel distributed control strategy that combines Bernoulli Thompson sampling [1] and Lloyd’s algorithm [2] to allow robots to actively search for and track targets, respectively. Compared to previous approaches that use only Lloyd’s algorithm, we show that the proposed algorithm significantly accelerates the speed at which robots find targets, particularly when large portions of targets congregate in some fixed areas in the environment.

I. CONTRIBUTIONS

In this work, we develop a novel control policy that enables robots to actively search for and track targets. We have three primary contributions: 1) we introduce a distributed active search algorithm based on dynamic TS, 2) we combine the TS-based search with Lloyd’s algorithm for active tracking, 3) we propose a goal swapping algorithm to more effectively assign goals to each robot, and 4) we demonstrate in a series of simulated experiments that a team of robots using the combined TS and Lloyd’s algorithms more effectively finds and tracks targets than a team that uses only Lloyd’s algorithm.

II. RESULTS

A. Simulation Environment

We test our proposed algorithms via MATLAB simulations. The task space is a 100 m × 100 m square. Targets may either be static or moving within their sub-regions at a maximum speed of 3 m/s. Existing targets may disappear (i.e., leave the environment) and new targets may appear (i.e., enter the environment) so that the number of targets may change over time.

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 10 m/s and are equipped with isotropic sensors with a sensing radius \( \rho_f = 5 \) m.

We use probability hypothesis density (PHD) filter for target state estimation and assume the robots do not have any prior knowledge of the targets. Thus, 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. We use the same measurement model for sensors as [3], with the exception of assuming that sensors are all as \([3]\), with the exception of assuming that sensors are all

B. Qualitative Comparison

We first show how active search using TS qualitatively improves multi-target tracking using a single trial. There are 40 robots searching for 40 targets, where 30 targets are located in a 33 m × 33 m square sub-region at the lower-left corner of \( E \), and another 10 targets in a 33 m × 33 m square sub-region at the top-right corner. We use PHD filter.

Figure [1] shows the locations of robots and targets at various points during exploration using both our previous [3] and new methods. When using our previous method, which only used Lloyd’s algorithm, a large portion of robots are idle even when a fair amount of targets are not tracked after 40 s, as the centroids of these orange diamonds are not located in any of the green circles in Fig. [1a]. After 60 s, as shown in Fig. [1b], robots tend to move towards the two corners with targets but a large portion of them have still not found any targets. We also see that robots get stuck on the boundary of target clusters, never reaching the interior targets. The result demonstrates the weakness of pure Lloyd’s 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 untracked targets.

On the other hand, the team using distributed Thompson sampling is able to quickly learn the target distribution and cluster in regions likely to contain targets. As Fig. [1c] shows, after 40 s a large number of the robots have already gathered at the large cluster of targets while a handful of other robots continue to search unexplored areas. After 60 s, most of the robots have found a target while a few continue to maintain coverage of these unlikely regions, as Fig. [1d] shows. The resulting emergent robot distribution is consistent with the target distribution, as the lower-left corner contains a higher number of robots than the upper-right corner (28 vs. 9, which closely match the number of targets in each region), while 3 robots monitor the rest of the area. This helps the team to track the targets more quickly as the more individuals are needed in an area, the more likely it is that an individual will be assigned to that region. Figure [1c] reflects the value of \( \alpha \) for each sampling candidates after 60 s, showing that the team has received more reward in areas with higher target concentrations.

C. Quantitative Comparison

To quantify the improvement in performance, we will use the first order Optimal SubPattern Assignment (OSPA)
Fig. 1. Figures show comparison of applying pure Lloyd’s algorithm and a combined Lloyd’s algorithm with Thompson sampling after 100 s. In Fig 1a, 1b, 1c, and 1d, green squares and circles show robot locations and sensor footprints, respectively. Orange diamonds show the locations of targets. Fig 1e maps $\alpha$ values in Fig 1d by darkness, with a darker color indicating a higher value.

Fig. 2. Boxplots show median OSPA errors and the 95% rise time of different teams of robots tracking static and dynamic targets using pure Lloyd’s algorithm and combination of Lloyd’s and TS.

Fig. 2

(a) OSPA Static
(b) OSPA Dynamic
(c) Rise Time Static
(d) Rise Time Dynamic

metric [4], a commonly-used approach in MTT. 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. We report the median OSPA value over the final 150 s of each trial, allowing the team to reach a steady state and smoothing out the effects of spurious measurements that cause the OSPA to fluctuate. We also show the 95% rise time of the OSPA error metric, i.e., the time it takes for the OSPA error to reach a value within 5% of the final value, to measure the speed at which robots reach steady state.

1) Tests Design: We test a range of team sizes (from 10 to 40 robots) and both search strategies (from [3] and the new method). For each configuration (environment, target type, team size) we run 10 trials, with the results aggregated into boxplots showing the steady state OSPA (to measure accuracy) and the 95% rise time (to measure speed).

2) Results: Figure 2 shows the results. As we see in Fig. 2a and Fig. 2b, the median OSPA decreases as the number of robots increases. This agrees with intuition as more robots should be able to better locate targets and there should be diminishing rewards with each added robot, i.e., going from 10 to 15 robots is more significant than 35 to 40 robots. We see that for static targets (Fig. 2a and Fig. 2c), our proposed method shows significantly lower OSPA error and rise time and that the variation of both parameters across trials is smaller, meaning it more accurate, faster, and more reliable. For dynamic targets (Fig. 2b and Fig. 2d), the OSPA error of teams using our proposed method are comparable or slightly higher and exhibit slightly more variation across trials, though neither effect is significant. Like the static case, our new method decreases both the magnitude and variation of the rise time, meaning it faster and more repeatable. We hypothesize that these differences in behavior are due to the ability of robots to actively sample the environment using coarse global information.

III. CONCLUSIONS

We develop a distributed control algorithm combining a novel distributed Thompson sampling algorithm with Lloyd’s algorithm to allow robots to effectively search for and track multiple moving targets without having any prior knowledge about targets. In particular, we see that the addition of TS allows robots to share coarse global information about recently detected targets in an efficient and scalable manner. As a result, teams using TS are able to more accurately track (stationary) targets, locate targets more quickly, and increase the consistency in performance. These trends are more pronounced in situations where targets are unevenly distributed within the search space.

REFERENCES