Communication-Aware Multi-robot Coordination with Submodular Maximization

Guangyao Shi[†], Ishat E Rabban[††], Lifeng Zhou[‡], Pratap Tokekar[††]

Abstract— Submodular maximization has been widely used in many multi-robot task planning problems including information gathering, exploration, and target tracking. However, the interplay between submodular maximization and communication is rarely explored in the multi-robot setting. In many cases, maximizing the submodular objective may drive the robots in a way so as to disconnect the communication network. Driven by such observations, in this paper, we consider the problem of maximizing submodular function with connectivity constraints. Specifically, we propose a problem called Communication-aware Submodular Maximization (CSM), in which communication maintenance and submodular maximization are jointly considered in the decision-making process. One heuristic algorithm that consists of two stages, i.e. *topology generation* and *deviation minimization* is proposed. We validate the formulation and algorithm through numerical simulation. We find that our algorithm on average suffers only slightly performance decrease compared to the pure greedy strategy.

I. INTRODUCTION

Many multi-robot cooperative tasks such as information gathering, exploration, target tracking can be formulated as submodular maximization problems [1]–[5]. The objectives in such problems (e.g., mutual information, area explored, number of targets tracked, etc.) have diminishing returns property which is shown to be submodular. Intuitively, submodularity formalizes the notion that adding more robots to a larger multi-robot team cannot yield a larger marginal gain in the objective than adding the same robot to a smaller team. Despite the fact that the submodular maximization problem is NP-hard, a simple greedy approximation algorithm can achieve nearly optimal performance [6], [7].

Communication plays a key role in successfully executing the greedy algorithm for a team of robots to choose actions. In the centralized case, robots may need to transmit their acquired information to either a leader of the team or a remote server; in the distributed case, robots may need to exchange local information with their neighbors to reach some global consensus [8], [9]. In either case, a team of robots needs to form a connected communication network for ensuring information flow. However, it is possible that actions that maximize the objective may lead the network to become disconnected. In fact, we expect that to happen quite often given that many of these objectives relate to coverage
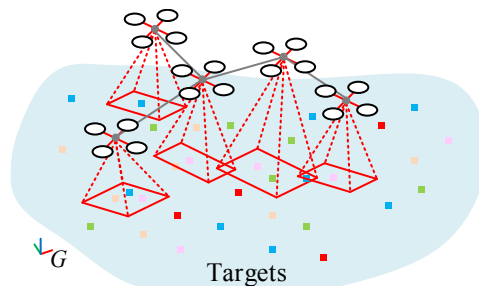

Fig. 1. One motivating example of this paper: multi-robot active target tracking with communication constraints. The team should maximize the number of targets observed at each planning step and and keep connected. Colored squares represent targets. Grey dots and lines represent induced communication network.

where the robots may want to move away from each other to reduce overlaps. Therefore, there is a need to introduce a connectivity constraint during decision-making. However, in the existing literature related to submodular maximization, communication is usually neglected, which motivates us to think about whether we can jointly consider communication maintenance and submodular maximization in the decision-making process for the team.

To this end, we propose a problem named Communication-aware Submodular Maximization (CSM) for a class of multi-robot task planning. In the proposed formulation, each robot needs to find one trajectory to be executed within the current planning epoch to maximize the submodular team objective. We also impose the constraint that the trajectories that are found must be such that the robots form a connected communication network at the end of the planning epoch. That is, the end positions of each of the trajectories that must form a connected network. We allow the robots to temporarily disconnect during the epoch. However, by ensuring connectivity at the end of the epoch, the robots will be able to exchange information gained during the epoch and jointly plan for the next epoch.

One motivating example is given in Fig. 1, in which a team of aerial robots equipped with downward-facing cameras are tracking targets on the ground. The objective is to maximize the number the targets observed within each epoch. On the one hand, robots need to move away from each other to reduce the overlap of the sensor footprints. On the other hand, they cannot move too far from each other while still ensuring a connected network at the end. A good solution for CSM is able to balance these two conflicting goals. In this paper, we propose a heuristic algorithm consisting of

[†]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA email:gyshi@terpmail.umd.edu.

[††]Department of Computer Science, University of Maryland, College Park, MD 20742 USA email: tokekar@umd.edu.

[‡]GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, USA e-mail: lfzhou@seas.upenn.edu. The author was with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA when part of the work was completed.

two stages, *topology generation* and *deviation minimization*, to solve CSM. The key idea of the proposed algorithm is that first, for each robot we discretize the problem by generating a set of candidate trajectories whose endpoints are within the reachable set of the robot and let robots choose trajectories greedily without considering communication constraints; then we make them minimally deviate from the endpoints corresponding to greedy selections to build connectivity. Specifically, in the *topology generation* stage, an edge-weighted graph is generated using robots' greedy selections, whose edge weights are defined over distance between pairs of robots. Then a Minimum Spanning Tree (MST) of the edge-weighted graph is extracted as the communication graph for the next epoch. In the *deviation minimization* stage, a Quadratic Programming (QP) is formulated to find new positions within the reachable set that minimally deviate from the greedy selections for robots to realize the MST in the workspace. We carry out extensive simulations to evaluate the performance of the proposed algorithm.

II. RELATED WORK

Submodular maximization and its variants have been widely used in multi-robot decision-making problems including coverage that [10]–[12], target tracking [1], [13]–[15], exploration [2], and information gathering [4], [7], [16]–[18]. These studies all based on the fact that greedy algorithm and its variants can solve submodular maximization problem its variants efficiently with provable performance guarantee. a However, communication is seldom considered in the existing works from robotic researchers and most of them assume that conna ected network is always there. Krause [18] uses the same assumption but associates some cost to edges of network and jointly optimizes coverage and network costs. But their work is static in nature and decision is made only once instead of at each planning epoch without partition matroid constraint [16]. By contrast, we consider the scenario where robots are moving and decisions need to be made at each planning epoch to maximize objective and guarantee connected network. Gharesifard [19] considers the submodular maximization case where one decision-maker represented as one vertex in the graph can access only the decisions of its neighbor and analyzes the influence of graph topology. [20] considers a similar problem but mainly from the perspective of a system designer. However, the graph discussed in [19], [20] is more like a relational graph instead of a communication graph, and authors all assume that the actions or decisions of vertices will not change the graph properties, i.e., connectivity. Grimsman [3] considers submodular maximization under topology constraints for multi-robot task allocation problems. If the topology constraints can be described as matroid constraints, a greedy algorithm can have a constant approximation factor. However, connectivity constraint is in general not a matroid constraint for robot team with the same communication radius, and authors in [3] make extra assumptions on task allocation structures to make spanning tree constraint a matroid constraint, which makes it unsuitable for our problem.

Another line of related work is on connectivity maintenance based on algebraic graph theory [21]–[24]. In these studies, connectivity maintenance and task are usually separately considered. Even though connectivity can be formally guaranteed when the task is not considered, complex task behaviors of robots in practice can break the connectivity. Our work is different from these works in two aspects: first, our formulation is rooted in combinatorial optimization, i.e., submodular maximization rather than continuous optimization and aims at solving discrete decision-making problems. Even though submodular set functions can be extended to continuous functions [25], the evaluation of the extended function involves exponentially many subsets of the ground set, which makes it unsuitable for robotic application. Second, we jointly consider team task and communication connectivity and therefore task behaviors will not break connectivity.

This work is also closely related to submodular maximization over graphs [26]–[28]. [28] considers the problem of finding a rooted arborescence in a directed graph with a budget to maximize a submodular function while by contrast graph considered in this paper is undirected and budget is not a constraint. In [26], authors consider the problem of finding a connected subgraph to maximize covered area, which is a special submodular set function, in the Euclidean plane and give one 2-approximation the algorithm. But their algorithm cannot be used to maximize general submodular functions that are interesting to multi-robot applications, for example mutual information [2], [5] and number of targets [1] because the performance of their algorithm relies on some additive properties of covered areas. [27] considers a similar problem as with [26] for general submodular functions. However, none of algorithms proposed in [26]–[28] can deal with partition matroid constraints [2], which is one inherent nature in multi-robot application since each robot has its unique set of choices. As a result, corresponding algorithms are not applicable to multi-robot applications.

Contribution: the main contribution of this paper is that we formulate a novel submodular maximization problem with connectivity constraints for multi-robot applications. A heuristic algorithm consisting of two stages is proposed. The correctness of the formulation and the proposed algorithm is validated through a case study on active target tracking.

III. PROBLEM FORMULATION

A. Preliminary

We first introduce some notations we will use in this paper. Given a set $\mathcal{A}$, $2^{\mathcal{A}}$ denotes the power set of $\mathcal{A}$. Given another set $\mathcal{B}$, the set $\mathcal{A} \setminus \mathcal{B}$ denotes the set of elements in $\mathcal{A}$ but not in $\mathcal{B}$. Given a set $\mathcal{V}$, a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$, and an element $v \in \mathcal{V}$, $f(v)$ is a shorthand that denotes $f(\{v\})$. We use $\Delta_f(r \mid \mathcal{A}) = f(\mathcal{A} \cup \{r\}) - f(r)$ to denote the marginal gain of adding r in $\mathcal{A}$. Whenever f is clear from the context, we will use shorthand $\Delta(r \mid \mathcal{A})$ for $\Delta_f(r \mid \mathcal{A})$. $[N]$ denotes the set $\{1, 2, \ldots, N\}$.

Definition 1 (Minimum Spanning Tree). *A minimum spanning tree (MST) is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.*

Definition 2 (Minimum Bottleneck Spanning Tree). *A minimum bottleneck spanning tree (MBST) in an undirected graph is a spanning tree in which the most expensive edge is as cheap as possible.*

B. System model

Suppose we have $N \geq 2$ robots in the environment, whose joint states at the start of the current epoch are denoted as $\mathbf{x} = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{dN}$, where d is the dimensionality and can be 2 or 3. Communication graph at the start of the current epoch $G = (V, E)$ is specified using proximity graph, i.e., a robot corresponds to a vertex $i \in V$ and $(i, j) \in E$ if $\|x_i - x_j\| \leq r_c$, where r_c is the communication radius.

Each robot is equipped with a sensor that gathers information based on the sensor footprint. For example, an aerial robot equipped with a downward-facing camera can sense the targets on the ground as shown in Fig. 1. In this paper, specifically, our goal is to find a set of trajectories, one per robot, to ensure that we maximize the submodular team objective within each epoch and ensure a connected network at the end of each epoch.

Let $\mathcal{R}_i$ be the reachable set of robot i of the current epoch, i.e., a set of locations that robot i can reach at the end of the current epoch from the start position of the current epoch. In general, there is no closed-form expression for the reachable set. In this paper, we assume that the reachable set can be approximately represented as obstacle-free convex polyhedrons or ellipsoid using an iterative optimization method such as the one presented in [29]. Let $\mathcal{T}_i$ be the set of trajectories of robot i each of which is within the reachable set $\mathcal{R}_i$. It should be noted that there are infinitely many elements in $\mathcal{T}_i$. Let $\mathcal{P}$ be the operator that can extract the endpoints of trajectories. With the above notations, the problem can be formulated as follows.

Problem 1 (Communication-aware Submodular Maximization). *At current planning epoch, robots' positions induce a connected communication graph G_e, we want to preserve the connectivity in the next epoch, i.e., induced graph G_{e+1} after moving to new positions is connected, meanwhile maximizing a submodular objective function. Mathematically,*

$$\max_{s_i \in \mathcal{T}_i, i \in [N]} f(\mathcal{S}) \tag{1}$$
$$s.t. \quad G_{e+1}(\mathcal{P}(\mathcal{S})) \text{ is connected,}$$

where $\mathcal{S} = \{s_1, \ldots, s_N\}$ and $\mathcal{T}_i$ is the set of trajectories for robot i.

One way to solve Problem 1 is by first discretizing the reachable set $\mathcal{R}_i$ into a set of discrete locations and by using lower-level motion primitives such that each point in the discretized reachability set is associated with one dynamically feasible trajectory [30].
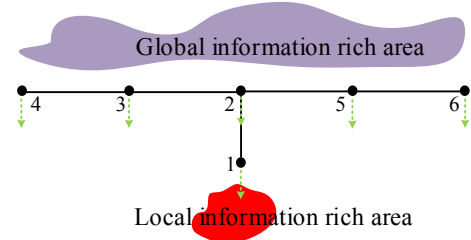

Fig. 2. One counter example to show that the sequential graph greedy strategy can be arbitrarily bad. If robot 1 is attracted to some local information rich area and firstly chooses an action to go downwards (dotted blue arrow), it may bias the whole team away from the global information rich area, which is on the top (other robots are forced to choose the blue arrows for connectivity).

Even after discretization, Problem 1 is still hard because the graph topology can be quite diverse and there can be exponentially many candidates. However, it should be noted that with proper discretization there is always a solution to Problem 1: since communication graph G_e is connected and all robots have non-empty reachable sets, one apparently feasible solution is that all robots do the same movement, e.g., the team as a whole shifts to a new position and G_{e+1} has the same graph topology as that of G_e.

One naive way to solve Problem 1 is to use a sequential greedy strategy and consider connectivity during the construction process as shown in Algorithm 1, in which at each iteration one robot that can connect to the existing connected graph will make decisions and the connected graph will then be correspondingly expanded. However, such a strategy can be arbitrarily bad as shown in Fig. 2. In the Sequential Graph Greedy (SGG) strategy, robot 1 may choose the action to go down first since it has the largest marginal gain w.r.t. empty graph, but his action will bias the team from more valuable area on the top. Another problem with such a strategy is that if robots have different reachable sets, SGG may end up with a disconnected graph. For example, in Fig. 2, if robot 1 has a large reachable set and move downwards to its best, other robots with a smaller reachable set may not be able to follow its pace to maintain the connectivity.

IV. HEURISTIC ALGORITHM

We propose a two-stage heuristic algorithm to solve Problem 1. The key idea is to first let robots greedily select trajectories, which of course may break the connectivity, and then make the robots minimally deviate from greedy selections to establish connectivity.

Topology Generation: In this stage, robots will greedily select trajectories based on marginal gains without considering the communication constraint. It is equivalent to solving a submodular maximization problem with partition matroid using a greedy strategy as shown in Algorithm 2. These greedy selections will virtually drive them to positions $x_1^g, \ldots, x_N^g$. Then we generate a complete graph K_N corresponding to these N robots. The edge weight between robots i and j is defined as $C(i, j) = \max(\frac{1}{2}(\|x_i - x_j\| - r_c), 0)$. If the distance between two robots is already within the
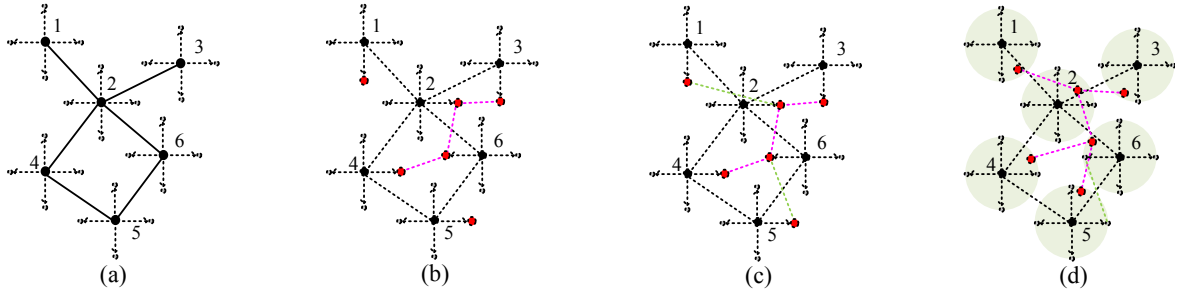
Fig. 3. One illustrative example of the proposed algorithm. (a) There are $N = 6$ robots in the team and each robot has five trajectories: go left, go right, stay, go up, and go down which corresponds to five dotted line in the work space. Current position of robots are denoted as black dots and the corresponding communication links are black lines. (b) The team greedily maximizes the submodular objective function which results in a disconnected graph i.e. robots 2, 3, 4, 6 are connected but robot 1 and 5 are disconnected from other robots. (c) Generate a complete weighted graph using greedy selections and then extract a MST from weighted graph K_6. Two blue edges are added. (d) Solve the deviation minimization problem to find the final positions for robots. The shaded circles centered at current positions are reachable sets for robots and the red graph is the network to be realized at next epoch.

Algorithm 1: Sequential Graph Greedy

---

1   <u>function SGG($f, \{\mathcal{T}_i\}_{i=1}^N, \mathcal{P}$)</u>

Input :
- A monotone submodular function f
- Partitioned ground set $\{\mathcal{T}_i\}_{i=1}^N$
- Operator $\mathcal{P}$ that can extract the end points of trajectories.

Output: A subset Sol of the ground set

2   $Sol \leftarrow \emptyset, \mathcal{T} \leftarrow \bigcup_{i=1}^N \mathcal{T}_i, G \leftarrow (V = \emptyset, E = \emptyset)$

3   **while** $|Sol| < N$ **do**

4   find the trajectory set $\mathcal{T}' \subseteq \mathcal{T}$ s.t. each element in $\mathcal{T}'$ can establish connections to the existing graph G at its endpoint

5   # find the element with largest marginal gain and its group ID

6      $a, \; i = \arg\max_{s \in \mathcal{T}'} \Delta f(s \mid Sol)$

7      $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_i, \quad Sol \leftarrow Sol \cup \{s\}$

8   G.add_node($\mathcal{P}(s)$),   G.add_edge($\mathcal{P}(s)$)

9   **end**

10   **return** Sol

Algorithm 2: Submodular Maximization with Partition Matroid

---

1   <u>function Greedy($f, \{\mathcal{T}_i\}_{i=1}^N, \mathcal{P}$)</u>

Input :
- A monotone submodular function f
- Partitioned ground set $\{\mathcal{T}_i\}_{i=1}^K$
- Operator $\mathcal{P}$ that can extract the end points of trajectories

Output: A subset Sol of the ground set

2   $Sol \leftarrow \emptyset, \mathcal{T} \leftarrow \bigcup_{i=1}^N \mathcal{T}_i$

3   **while** $|Sol| < N$ **do**

4   # find the element with largest marginal gain and its ID

5      $s, \; i = \arg\max_{s \in \mathcal{T}} \Delta(s \mid Sol)$

6      $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_i$

7      $Sol \leftarrow Sol \cup \{s\}$

8   **end**

9   **return** Sol

---

communication radius, then the cost of that edge will be zero. If the distance is greater than r_c, then at least one of the two robots needs to move at least $\frac{1}{2}(\|x_i - x_j\| - r_c)$ to realize the edge.

After defining the K_N, we use MST algorithm [31] to extract a connected graph for the next epoch. The intuition for using MST to generate graph topology is: first, the costs of edges are defined over distances to be traveled to realize edges. Therefore, the graph returned by MST algorithm to some extent reflects the total deviation to achieve connectivity; second, as shown in [31], MST is also a MBST, which suggests that the graph returned by MST algorithm to some extent also reflects the minimum distance to be deviated to achieve connectivity.

Deviation Minimization: After generating the graph topology, in this stage we consider the problem of how to make robots minimally deviate from the greedy selections to establish connectivity. The problem is described below.

Problem 2 (Deviation Minimization). *Given current positions* $x_1, \ldots, x_N$ *of robots, positions* $x_1^g, \ldots, x_N^g$ *corresponding to greedy selection, reachable sets* $\mathcal{R}_1, \ldots, \mathcal{R}_N$, *communication radius* r_c, *and a connected graph* G_{e+1} *to be realized, the goal is find new locations* $x_1^*, \ldots, x_N^*$ *such that*

- $x_1^*, \ldots, x_N^*$ *can realize all edges in graph* G_T, *which is returned by MST algorithm;*

- *The deviations of robots from greedily selected positions* $x_1^g, \ldots, x_N^g$ *are minimized.*

- $x_1^*, \ldots, x_n^*$ *are within reachable set w.r.t their current positions.*

- *Robots will not collide in the new positions.*

Mathematically,

$$\min \sum_{i=1}^{N} w_i \left\| x_i^* - x_i^g \right\|$$
$$s.t. \left\| x_i^* - x_j^* \right\| \le r_c, (i,j) \in G_T.edges \qquad (2)$$
$$x_i^* \in \mathcal{R}_i, i \in [N]$$
$$\left\| x_i^* - x_j^* \right\| > r_s, \forall i \ne j,$$

where $w_i \ge 0$ is parameter used to describe robot i's willingness to deviate from x_i^g and more details on choosing w_i are given in Sec. V and r_s is the safe radius for each robot.

It should be noted that the safety constraints are not convex but they can be well handled by available solves like Gurobi 9.0 [32].

One illustrative example is shown in Fig. 3 to demonstrate how our proposed algorithm works. In the formulation given above, the reachable set $\mathcal{R}_i$ can be an arbitrary set. We also have not defined w_i which can be chosen based on the application. In the following, we will describe one specific example on active target tracking in which we assume that the reachable set can be represented as a circle and present three ways to select w_i based on the number of targets that a robot can track during one epoch.

V. CASE STUDY: ACTIVE TARGET TRACKING

In this section, we present one case study on multi-robot active target tracking with communication constraints, in which each robot has a downward-facing camera and the team aims to maximize the number of targets that will be observed in each planning epoch. This case study is presented to demonstrate:

- Correctness: The proposed algorithm will return a solution that preserves connectivity. This will be validated by checking the network generated by the team after taking action.
- Performance: The proposed algorithm outperforms SGG strategy and has a competitive performance compared with a simple greedy strategy, which can be viewed as the empirical performance upper bound for the team at each planning epoch. We compare our algorithm with these two algorithms with respect to the number of targets tracked.

A. System Model

Target Model: We assume each target has single integrator motion model,

$$p^j(t+1) = p^j(t) + v^j(t),$$

where $p^j(t)$ and $v^j(t)$ denote the position and the velocity of target $j = 1, \ldots, 50$. The robot obtains noisy measurements of the targets' positions and use a Kalman filter for estimation. The noise is set to Gaussian noise with zero mean and 0.5 standard deviation. Each target's velocity is
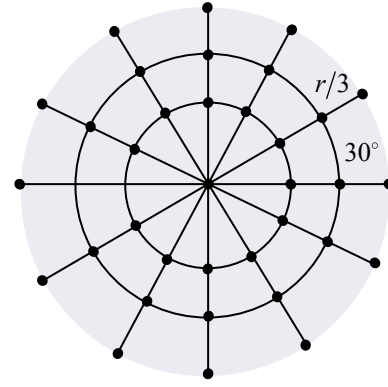

Fig. 4. One illustrative example to demonstrate the discretization of the reachable set. Endpoints are uniformly sampled w.r.t radius (step size $\frac{r}{3}$) and angles (step size $30°$)   .

initialized to be zero and is updated by using two consecutive measurements and the time interval of these two measures:

$$v^j(t') = \frac{\tilde{p}^j(t') - \tilde{p}^j(t)}{t' - t},$$

where $\tilde{p}^j(t')$ and $\tilde{p}^j(t)$ are two measures at time step t' and t with $t' > t$.

Robot Model: We assume that all aerial robots fly at fixed heights and the reachable set of each robot can be described as a circle centered at its current position with a radius 4 meters. Then the reachable set is discretized w.r.t. both the radius and the angle as shown in Fig. 4 to generate trajectory set. Each robot is equipped with a downward-facing camera and is able to observe ground targets inside the sensor footprints. We conduct 10 rounds of simulation and in each simulation there are 10 epochs. Communication radius is set to be 10 meters.

Weight Selection: in this paper we test two ways to set parameter w_i, which is about robot i's willingness to deviate from its endpoints, for Problem 2.

1) w_i is set to be the individual gain of each robot, i.e.,

$$w_i = f(s_i),$$

where s_i is the trajectory for robot i selected using greedy algorithm. We will refer this approach as *Weight1* in the Sec. V-B. The intuition here is that if a robot itself alone can observe many targets, it is less willing to deviate from its selection.

2) w_i is set to be marginal gain of robot i's selected trajectory $s_i \in \mathcal{T}_i$, i.e.,

$$w_i = f(\mathcal{S}) - f(\mathcal{S} \setminus s_i),$$

where $\mathcal{S}$ is a set of trajectories selected using greedy strategy. We will refer this approach as *Weight2* in the Sec. V-B. The intuition here is that the willingness of a robot to deviate from the greedy selection is reflected in its contribution to the whole team. The more it contributes to the team, i.e., large marginal gain, the less willing it is to deviate from its selection.
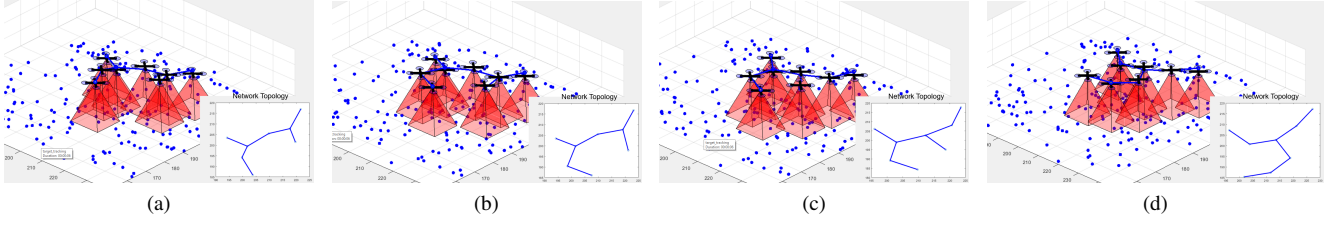
Fig. 5. Screenshots of target tracking process. There are eight robots tracking moving targets. The bottom right small graph in each figure show the network topology of the team at the start of that epoch. Sub-figures (a)-(c) show how the topology evolves over time.

3) Another way to set w_i is to consider the variation of objective values around the point x_i^g. Specifically, given the position x_i^g of robot i corresponding to its greedy selection, we uniformly sample a set $\{x_1^i, \ldots, x_W^i \mid \left\| x_j^i - x_i^g \right\| \leq 1, j = 1, \ldots, W\}$ and define the weight for robot i as

$$w_i = f(x_i^g) - \min_{x \in \{x_1^i, \ldots, x_W^i\}} f(x).$$

We will refer this approach as *Weight3* in the Sec. V-B. The intuition here is that a robot will check the drop of the number of targets observed if he deviates from its selection. If the drop is large, then it is less willing to deviate from its selection.

It should be noted that after setting w_i to be some non-negative constant, Problem 2 becomes a standard QP problem and can be solved using commercial solvers such as Gurobi.

B. Simulation Results

All experiments were performed on a Windows 64-bit laptop with 16 GB RAM and an 8-core Intel i5-8250U 1.6GHz CPU using MATLAB with Gurobi 9.0 [32], which can deal with both convex and non-convex quadratic constraints. We test three algorithms in this section: the proposed algorithm, the greedy algorithm without considering communication constraints, and SGG. Here the greedy algorithm corresponds to the greedy selection at the beginning of each epoch without considering communication constraints, which can be viewed as an empirical upper bound of the team performance at each epoch. We compare with such results to show that the deviation minimization part of the proposed algorithm will only slightly reduce the performance of the team from the upper bound. The data on SGG is collected in the following way: we initialize the robots and targets in the same positions as that in the test setup of the proposed algorithm and run SGG to track targets. We use the results of SGG as a baseline.

As shown in Fig. 6, the proposed algorithm can achieve the 90% of the performance of the pure greedy strategy. Moreover, the proposed algorithm on average can track ten percent more targets compared to SGG. As for three ways to select weights in deviation minimization problem, they have similar performances. Readers are referred to multimedia submission for an animation that shows the connectivity during operation of the team and one screenshot is shown in Fig. 5.
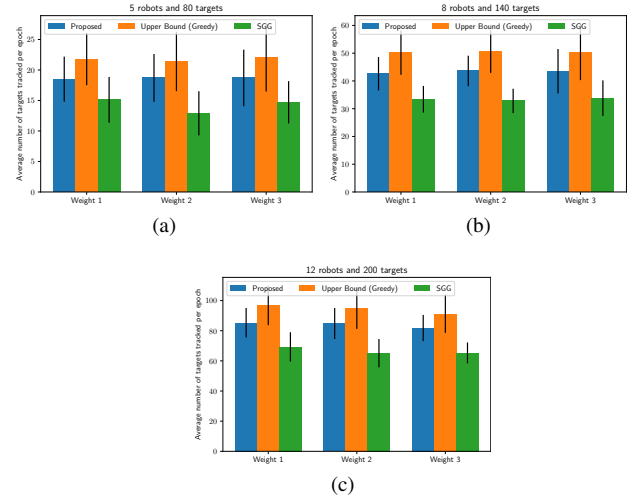

Fig. 6. Average number of targets observed over each epoch. (a) There are 5 robots and 80 targets. (b) There are 8 robots and 140 targets. (c) There are 12 robots and 200 targets. The proposed algorithm, greedy algorithm without considering communication at each epoch, and SGG are denoted in blue, orange, and greed bars respectively. The black line above bars represent one standard deviation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a problem named Communication-aware Submodular Maximization (CSM) for a class of multi-robot task planning and propose a heuristic algorithm consisting of two stages, topology generation and deviation minimization, to solve CSM. There are two directions to extend this work. One is to come up with bounded approximation algorithm to solve CSM or analyze the conditions under which some performance guarantees can be claimed. Another direction is to find efficient algorithms to approximate and discretize the reachable set in the cluttered environment.

REFERENCES

[1] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 129–136, 2018.
[2] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Autonomous Robots*, vol. 43, no. 2, pp. 485–501, 2019.
[3] R. K. Williams, A. Gasparri, and G. Ulivi, "Decentralized matroid optimization for topology constraints in multi-robot allocation problems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 293–300.

[4] S. Jorgensen, R. H. Chen, M. B. Milam, and M. Pavone, "The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5622–5629.

[5] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[6] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[7] A. Krause and C. Guestrin, "Submodularity and its applications in optimized information gathering," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 4, pp. 1–20, 2011.

[8] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.

[9] E. Cristofalo, E. Montijano, and M. Schwager, "Consensus-based distributed 3d pose estimation with noisy relative measurements," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 2646–2653.

[10] X. Sun, C. G. Cassandras, and X. Meng, "Exploiting submodularity to quantify near-optimality in multi-agent coverage problems," *Automatica*, vol. 100, pp. 349–359, 2019.

[11] R. K. Ramachandran, L. Zhou, and G. S. Sukhatme, "Resilient coverage: Exploring the local-to-global trade-off," *arXiv preprint arXiv:1910.01917*, 2019.

[12] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[13] L. Zhou and P. Tokekar, "Sensor assignment algorithms to improve observability while tracking targets," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1206–1219, 2019.

[14] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Distributed attack-robust submodular maximization for multi-robot planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2479–2485.

[15] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3067–3072.

[16] B. Schlotfeldt, V. Tzoumas, D. Thakur, and G. J. Pappas, "Resilient active information gathering with mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4309–4316.

[17] G. Shi, L. Zhou, and P. Tokekar, "Robust multiple-path orienteering problem: Securing against adversarial attacks," in *Robotics: Science and Systems (RSS)*, 2020.

[18] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proceedings of the 5th international conference on Information processing in sensor networks*, 2006, pp. 2–10.

[19] B. Gharesifard and S. L. Smith, "Distributed submodular maximization with limited information," *IEEE transactions on control of network systems*, vol. 5, no. 4, pp. 1635–1645, 2017.

[20] D. Grimsman, M. S. Ali, J. P. Hespanha, and J. R. Marden, "The impact of information in greedy submodular maximization," *IEEE Transactions on Control of Network Systems*, 2018.

[21] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.

[22] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 2919–2924.

[23] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[24] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[25] F. Bach, "Learning with submodular functions: A convex optimization perspective," *arXiv preprint arXiv:1111.6453*, 2011.

[26] C.-C. Huang, M. Mari, C. Mathieu, J. S. Mitchell, and N. H. Mustafa, "Maximizing covered area in the euclidean plane with connectivity constraint," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[27] T.-W. Kuo, K. C.-J. Lin, and M.-J. Tsai, "Maximizing submodular set function with connectivity constraint: Theory and application to networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 533–546, 2014.

[28] R. Ghuge and V. Nagarajan, "Quasi-polynomial algorithms for submodular tree orienteering and other directed network design problems," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 1039–1048.

[29] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 109–124.

[30] M. S. Branicky, R. A. Knepper, and J. J. Kuffner, "Path and trajectory diversity: Theory and algorithms," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1359–1364.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[32] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com