An Approximation Algorithm for Distributed Resilient Submodular Maximization

Lifeng Zhou and Pratap Tokekar

Abstract— **We study a distributed resilient submodular maximization problem in which a group of robots collaboratively choose a strategy set. The global objective is to maximize a submodular function on the strategy set with the existence of a known number of robot attacks or failures. When choosing a strategy, each robot communicates with other robots within a local group, due to its limited communication ability. For such problem, we propose a distributed resilient submodular maximization algorithm that takes into account both the limited information available for the robots and the attacks or failures. In particular, our algorithm guarantees an approximation performance that is within a constant factor of the optimal strategy. Our analysis resorts to the curvature of the submodular set function, and proves that the algorithm is scalable, runs in polynomial time and is faster than its centralized communication manner. We demonstrate the efficacy of our algorithm through both Matlab and Gazebo simulation with a multi-robot target tracking scenario.**

I. Introduction

Resiliency is a hot topic across both academia [1], [2], [3] and industry [4], [5], [6]. With resiliency, we take the view that cyber attacks are unavoidable. As such, some part of the system is likely to be compromised. What we would like is to ensure the overall system continues to perform at an acceptable level despite these compromised assets. Motivated by this goal, researchers have developed algorithms for improving the resiliency of the system in a variety of areas such as smart grid and power systems [1], [7], IT data and infrastructure protection [5], [6], medical monitoring [8], control systems [2], [9] and robotics [3], [10]. Typical examples include:

- *(Power system)* How to maintain the acceptable levels of operation for a power generator in the face of sudden faults or attacks [7]?
- *(Medical monitoring)* How to acquire accurate detection of medical conditions subject to external perturbations and internal faults [8]?
- *(Estimation and control)* How to reconstruct the state of the linear system and stabilize it in the presence of sensor or actuator attacks [2]?

In this paper, we focus on the resiliency in multi-robot systems where robots interact locally with their nearest neighbors to collaboratively achieve certain goals against attacks or failures [1]. For example, Saulnier, et al. proposed a distributed resilient formation control approach to achieve the flocking behavior of multiple robots in the presence of defective or malicious robots [3]. Similarly, when some of the robots are non-cooperative, Saldana, et al. provided a distributed resilient strategy to steer the robots to achieve consensus with time-varying communication graphs [11]. Following this line, Guerrero-Bonilla, et al. presented sufficient conditions to guarantee resilient consensus on triangular and square lattices by studying the robot communication range [12].

Unlike the goal to achieve desired formations of robots, we are interested in maximizing a common objective function defined on the strategy set collaboratively selected by the robots. Examples of the objective function can be the joint area covered by the robots/sensors in the environment [13] and the total number of targets detected by the robots at a prescribed time [14], [10]. This kind of objective function usually turns out to be a submodular function. Submodularity indicates a function with the property of diminishing returns. Many information-based measures, such as entropy and mutual information [13], and geometric measures such as the visibility area [15], are known to be submodular. Maximizing the submodular function is generally NP-complete. However, a simple greedy algorithm yields a constant-factor approximation guarantee by selecting a strategy with the maximum marginal contribution to the function value in each round [16], [17].

We take into account the similar settings of attacks and the distributed communication manner as presented in the multi-robot formation studies [3], [11], [12]. In particular, we consider that robots can fail or its sensors can get attacked [18], and the robot has the limited sensing and communication range so that it can make decisions based on the local information only [19], [20]. In fact, when there exists a given number of worst-case attacks, the standard greedy algorithm loses its approximation guarantee and can do arbitrarily bad [21]. Besides, if only local information is provided, the standard greedy algorithm cannot be directly implemented since it requires the decisions from the entire network [19], [20]. Thus, in this paper, we tend to address a submodular maximization problem with both distributed communication and robot failures.

Related work. *(Resilient submodular maximization)* Tzoumas, et al. proposed a resilient submodular maximization algorithm by combining an oblivious approach ("select strategy with the largest contribution without considering the redundancy") with the standard greedy algorithm [22], [21]. They show the resilient algorithm has a provable performance guarantee (close to optimal) to play against a given

The authors are with the Department of Electrical and Computer Engineering, Virginia Tech USA. {lfzhou, tokekar}@vt.edu.

This material is based upon work supported by the National Science Foundation under Grant No. 479615.

[1]Henceforth, we refer to "attack" and "failure" interchangeably.

number of worst-case attacks, and it runs as fast as the standard greedy algorithm. Building on this work, Schlotfeldt, et al. extended the resilient submodular maximization algorithm to the multi-robot information gathering [23] and we studied a resilient multi-robot target tracking problem [10].

(Distributed submodular maximization) When only local information is available, Gharesifard and Smith proposed a distributed submodular maximization approach where each agent sequentially takes a greedy action based on the decisions from its previous neighbors [19]. They show this sequential greedy approach achieves a constant-factor approximation guarantee that is the same as that of the standard greedy algorithm with global/centralized information. Following this line, Grimsman, et al. improved the approximation performance of the sequential greedy algorithm by studying the structure of the communication graph [20]. In these two works, even though agents make decisions by only considering the decisions from their previous neighbors, they need to perform sequentially, which makes the algorithm inefficient when the number of robots is large. Another branch for solving distributed submodular maximization is to deal with large-scale machine learning problems which require selecting a representative subset out of a massive data set. For finding the representative subset, Mirzasoleiman, et al. proposed a two-stage greedy algorithm [24]. In the first stage, they partition the massive data into many small subsets and then run a standard greedy algorithm on these subsets in parallel to obtain a local solution on each subset. In the second stage, they collect these local solutions and run the standard greedy algorithm again on a central processor to select out the representative subset. They show this two-stage greedy approach has a provable performance guarantee and performs efficiently on the sparse Gaussian process inference and exemplar-based clustering with tens of millions of data points.

In this paper, we handle these two challenges i.e., attacks and local communication, simultaneously in a submodular maximization problem where a group of robots makes collaborative decisions. The robots can only communicate locally due to the limited communication range. They collaboratively select a set of strategies to maximize a common submodular objective function against a known number of the worst-case attacks. Inspired by the "*partition*" stage of the distributed greedy approach in [24], we let each robot first identify a local unique group it belongs to based on the limited communication range. In this way, the whole robot network can be partitioned into smaller separated subgroups. Then the robots within the same subgroup collaboratively take a resilient approach to play against the worst-case attacks [21], [10], ignoring the strategies from other groups. With this approach, all cliques of robots can perform in parallel.

Contributions. We make the following contributions:

- (*Problem*) We formalize the problem of distributed resilient submodular maximization against the *worst-case attacks* by communicating within the *local group* only.
- (*Solution*) We propose the first algorithm for such problem, and prove it has the following properties:
  - *provable approximation performance*: the algorithm ensures a constant-factor approximation performance of the optimal for any objective function that is monotone and submodular;
  - *minimal running time*: the algorithm is scalable and runs in polynomial-time. It is faster than the centralized communication, especially when the communication graph is sparse;
- (*Empirical Evaluation*) We illustrate the performance for resilient target tracking against robot attacks, and the efficacy of our approach through both Matlab and Gazebo simulations.

Overall, in this paper we go beyond the centralized resilient submodular maximization [22], [21], [23], [10] by proposing the distributed submodular maximization; and go beyond the distributed submodular maximization [24], [19], [20] by proposing the resilient submodular maximization.

Organization of rest of the paper. We formulate a distributed resilient submodular maximization problem in Section II. We present a distributed resilient submodular maximization algorithm in Section III along with the analysis of its approximation ratio and computational complexity in Section IV. We illustrate the performance of the proposed algorithm by Matlab and Gazebo simulations in Section V. We conclude the paper in Section VI.

Notations: Given a set $\mathcal{A}$, $2^{\mathcal{A}}$ denotes its power set; $|\mathcal{A}|$ denotes $\mathcal{A}$'s cardinality; given another set $\mathcal{B}$, the set $\mathcal{A} \setminus \mathcal{B}$ denotes the set of elements in $\mathcal{A}$ that are not in $\mathcal{B}$. A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. A clique in graph G is a subgraph of G that is complete. Denote $\mathcal{K}(G)$ as the number of the non-overlapping cliques in graph G. Denote $\omega(G)$ as the clique number of graph G, which is the number of vertices in the largest clique in graph G.

II. Problem Formulation

In this section, we propose a distributed resilient submodular maximization problem. We are given N robots on a communication graph $\mathcal{G}$ with nodes $\mathcal{R} = \{1, \cdots, N\}$. Each robot $i \in \mathcal{R}$ has a candidate *strategy set* $\mathcal{X}_i$. The robot must choose one strategy (action) $s_i \in \mathcal{X}_i$, which follows a partition matroidal constraint [17]. We assume the number of candidate strategies for each robot i, $|\mathcal{X}_i| = D,\ i \in \mathcal{R}$. We define a ground set of strategies $\mathcal{X} = \bigcup_i \mathcal{X}_i$ and are given a normalized, monotone (increasing) and submodular function, $f : 2^{\mathcal{X}} \to \mathbb{R}_{\geq 0}$. The function value of a strategy s_i is $f(\{s_i\})$ and its shorthand, $f(s_i)$. With a slight abuse of notation, we denote the set of robots with strategy set $\mathcal{S} = \bigcup_i s_i$ as $\mathcal{R}(\mathcal{S})$ and denote the strategy set for a set of robots $\mathcal{C}$ as $\mathcal{X}(\mathcal{C})$. Evidently, $\mathcal{R} = \mathcal{R}(\mathcal{X})$ and $\mathcal{X} = \mathcal{X}(\mathcal{R})$.

We consider robots choosing their strategies in a distributed communication manner. The robots can only communicate and share strategies within the same local group. We call this local group as the clique of robots on the graph

$\mathcal{G}$. And there is no communication allowed (or required) between cliques. We assume there exists a known number of the worst-case attacks to the whole robot team. We also assume that the number of attacks is less than the number of total robots. Each clique only knows the total number of the attacks and has no idea of how these attacks are distributed among cliques. The objective is to maximize a submodular function defined on the strategy set selected by each robot against the worst-case attacks. We propose the problem as below:

Problem 1 (Distributed Resilient Submodular Maximization).

$$\begin{aligned} \max_{\mathcal{S}\subseteq\mathcal{V},|\mathcal{S}|\le N} \quad \min_{\mathcal{A}\subseteq\mathcal{S},|\mathcal{A}|\le\alpha} \quad & f(\mathcal{S}\setminus\mathcal{A}): \\ & |\mathcal{S}\cap\mathcal{X}_i| = 1,\ \forall i\in\mathcal{R}; \\ \mathcal{S} = \mathcal{S}_1\cup\cdots\cup\mathcal{S}_{\mathcal{K}(\mathcal{G})},\ & |\mathcal{S}_k|\le n_k; \\ n_1+\cdots+n_{\mathcal{K}(\mathcal{G})} & = N; \\ |\mathcal{A}|\le\alpha,\ & \alpha<N, \end{aligned} \tag{1}$$

where $|\mathcal{S}\cap\mathcal{X}_i| = 1$ denotes a partition matroid constraint that each robot i must choose one strategy from its strategy set $\mathcal{X}_i$. $\mathcal{S}$ is the decision set selected by all the cliques of robots. $\mathcal{S}_k$ and n_k are the strategy set and the number of robots in each clique $k, k\in\{1,\cdots,\mathcal{K}(\mathcal{G})\}$. $\mathcal{A}$ denotes attack set from the selected set $\mathcal{S}$. The constraint $|\mathcal{A}|\le\alpha$ captures the problem assumption that at most α robots in the network can fail or get attacked.

Problem 1 can be interpreted as a two-stage perfect information sequential game [25, Chapter 4] between $\mathcal{K}(\mathcal{G})$ cliques on graph $\mathcal{G}$ and an attacker. The cliques first work in parallel to select out a strategy set $\mathcal{S}$ to maximize the objective function value. The strategy set $\mathcal{S}$ is the union on the strategy set $\mathcal{S}_k$ from each clique. By observing the strategy set $\mathcal{S}$, the attacker then executes a worst-case attack $\mathcal{A}$ from $\mathcal{S}$ to minimize the objective function value.

III. Algorithm

We present our main algorithm for solving Problem 1 in Algorithm 1. Since we assume that robots select strategies based on the information within the same clique, we first introduce related approaches to partition the communication graph into separated subgroups.

A. Clique Cover

We assume each robot has a limited communication range and can communicate with other robots within its communication range. We set the communication ranges of all the robots to be equal. Then the underlying communication topology of the robots is an undirected communication graph. Given this communication setup, we propose a distributed algorithm to partition the robot communication graph into separated cliques (in the appendix). Notably, this clique partition problem is also called "*non-overlapping clique cover*" in the literature, which is NP-hard even for a centralized solution [26]. Thus, we assume for a stationary communication graph, each robot knows its unique clique.

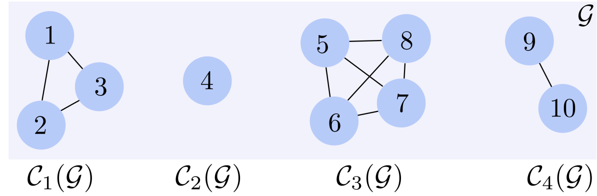


Fig. 1. A graph $\mathcal{G}$ contains 10 robots and 4 cliques.

If the communication is dynamic, each robot can identify its unique clique by our distributed clique cover algorithm or other related algorithms. After each robot identifies its unique clique, the robots together formulate an undirected communication graph $\mathcal{G}$ with non-overlapping cliques. We show an example of the communication network in Fig. 1 where graph $\mathcal{G}$ contains 10 robots and 4 cliques, $\mathcal{C}_1(\mathcal{G}),\cdots,\mathcal{C}_4(\mathcal{G})$. Notably, a clique can have a single robot, say $\mathcal{C}_2(\mathcal{G})$.

B. Distributed Resilient Submodular Maximization Algorithm

We then describe our distributed resilient submodular maximization algorithm for solving Problem 1 in Algorithm 1.

After the non-overlapping clique cover, all cliques of robots work in parallel to perform against the attacks (Alg 1, line 2). Notably, each clique of robots only knows the total number of attacks, α for the whole robot network $\mathcal{G}$. It does not know how α attacks are distributed among the cliques. Thus, each clique conjectures the worst-case scenario and makes the most conservative guessing. That is, each clique $\mathcal{C}_i(\mathcal{G})$ considers the number of attacks as α in it.

1) If the number of attacks α is less than its size (Alg. 1, line 3), it sets the number of attacks as α. Then a resilient algorithm is executed in two steps. First, the clique sequentially constructs a *local oblivious set* $\mathcal{S}_k^o$ by adding one strategy at a time from $\mathcal{X}(\mathcal{C}_k(\mathcal{G}))$ to $\mathcal{S}_k^o$ (Alg. 1, lines 4-9). Specifically, $\mathcal{S}_k^o$ is constructed such that it satisfies both the attack cardinality constraint (line 4) and the "*one strategy per robot*" constraint (line 6). Also, $\mathcal{S}_k^o$ is constructed such that each strategy $s\in\mathcal{X}(\mathcal{C}_k(\mathcal{G}))$ added in $\mathcal{S}_k^o$ achieves the highest value of $f(s)$ among all the strategies in $\mathcal{X}(\mathcal{C}_k(\mathcal{G}))$ that have not been yet added in $\mathcal{S}_k^o$ and can be added in $\mathcal{S}_k^o$ (Alg. 1, lines 4-7). Second, the remaining robots, $\mathcal{C}_k(\mathcal{G})\setminus\mathcal{R}(\mathcal{S}_k^o)$ whose strategies are not selected in the *local oblivious set* $\mathcal{S}_k^o$ sequentially construct a *local greedy set*, $\mathcal{S}_k^g$ (Alg. 1, lines 10-15). $\mathcal{S}_k^g$ is constructed by picking greedily a strategy from $\mathcal{X}(\mathcal{C}_k(\mathcal{G})\setminus\mathcal{R}(\mathcal{S}_k^o))$ at a time such that it satisfies "*one strategy per robot*" constraint (Alg. 1, line 12). Also, $\mathcal{S}_k^g$ is constructed such that each strategy $s\in\mathcal{X}(\mathcal{C}_k(\mathcal{G})\setminus\mathcal{R}(\mathcal{S}_k^o))$ added in $\mathcal{S}_k^g$ achieves the highest marginal contribution $f(\mathcal{S}_k^g\cup\{y\}) - f(\mathcal{S}_k^g)$ among all the strategies in $\mathcal{X}(\mathcal{C}_k(\mathcal{G})\setminus\mathcal{R}(\mathcal{S}_k^o))$ that have not been yet added in $\mathcal{S}_k^g$ and can be added in $\mathcal{S}_k^g$ (Alg. 1, lines 10-13).
2) If the number of attacks is larger than the clique's size (Alg. 1, line 16), the clique sets the number of

Algorithm 1: Distributed Resilient Submodular Maximization

Input:
- set of robots $\mathcal{R}$
- robot decision set $\mathcal{X}_i$, $\forall i \in \mathcal{R}$
- objective function f
- number of attacks α

Output: robots' strategy set $\mathcal{S}$

1: $\mathcal{S}_k \leftarrow \emptyset$; $\mathcal{S}_k^o \leftarrow \emptyset$; $\mathcal{S}_k^g \leftarrow \emptyset$; $k = \{1, \cdots \mathcal{K}(\mathcal{G})\}$
2: **for** each clique $\mathcal{C}_k(\mathcal{G})$ **do**
3:   **if** $\alpha < |\mathcal{C}_k(\mathcal{G})|$
4:     **while** $|\mathcal{S}_k^o| < \alpha$ **do**
5:       $s \in \arg\max_{y \in \mathcal{X}(\mathcal{C}_k(\mathcal{G}))} \ f(y)$
6:       **if** $|(\mathcal{S}_k^o \cup \{s\}) \cap \mathcal{X}_i| = 1, \ \forall i \in \mathcal{C}_k(\mathcal{G})$
7:         $\mathcal{S}_k^o \leftarrow \mathcal{S}_k^o \cup \{s\}$
8:       **end if**
9:     **end while**
10:     **while** $|\mathcal{S}_k^g| < |\mathcal{C}_k(\mathcal{G}) \setminus \mathcal{R}(\mathcal{S}_k^o)|$ **do**
11:       $s \in \arg\max_{y \in \mathcal{X}(\mathcal{C}_k(\mathcal{G}) \setminus \mathcal{R}(\mathcal{S}_k^o))}$ $f(\mathcal{S}_k^g \cup \{y\}) - f(\mathcal{S}_k^g)$
12:       **if** $|(\mathcal{S}_k^g \cup \{s\}) \cap \mathcal{X}_i| = 1, \ \forall i \in (\mathcal{C}_k(\mathcal{G}) \setminus \mathcal{R}(\mathcal{S}_k^o)$
13:         $\mathcal{S}_k^g \leftarrow \mathcal{S}_k^g \cup \{s\}$
14:       **end if**
15:     **end while**
16:   **else**
17:     **while** $|\mathcal{S}_k^o| < |\mathcal{C}_k(\mathcal{G})|$ **do**
18:       $s \in \arg\max_{y \in \mathcal{X}(\mathcal{C}_k(\mathcal{G}))} \ f(y)$
19:       **if** $|(\mathcal{S}_k^o \cup \{s\}) \cap \mathcal{X}_i| = 1, \ \forall i \in \mathcal{C}_k(\mathcal{G})$
20:         $\mathcal{S}_k^o \leftarrow \mathcal{S}_k^o \cup \{s\}$
21:       **end if**
22:     **end while**
23:     $\mathcal{S}_k^g \leftarrow \emptyset$
24:   **end if**
25:   $\mathcal{S}_k = \mathcal{S}_k^o \cup \mathcal{S}_k^g$
26: **end for**
27: $\mathcal{S} = \bigcup_{k=1}^{\mathcal{K}(\mathcal{G})} \mathcal{S}_k$

attacks as its size. Similarly, the clique constructs a *local oblivious set* $\mathcal{S}_k^o$ with cardinality as its size and satisfying "*one strategy per robot*" constraint (Alg. 1, lines 17-22). Evidently, there are no remaining robots whose strategies have not been selected in $\mathcal{S}_k^o$. Thus, the *local greedy set* $\mathcal{S}_k^g$ is set to be empty (Alg. 1, line 23).

3) The strategy set in this clique is the union set of the *local oblivious set* and the *local greedy set* (Alg. 1, line 25).

Overall, the strategy set selected by all the robots $i \in \mathcal{R}$ is the union set of the strategy sets from all the cliques (Alg. 1, line 27).

IV. Performance analysis

We quantify the performance of Algorithm 1, by bounding its approximation ratio and the running time. We describe the approximation performance by using the curvature $\nu_f(\mathcal{I})$ of

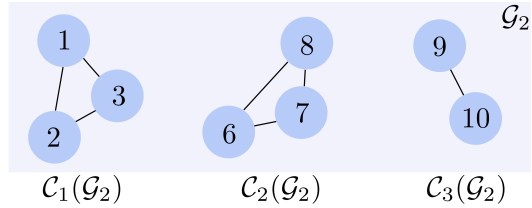


Fig. 2. Subgraph $\mathcal{G}_2$.

the submodular function f, the *global oblivious set* and a subgraph of graph $\mathcal{G}$.

Curvature [27]: consider a matroid $\mathcal{I}$ for ground $\mathcal{X}$, and a non-decreasing submodular set function $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ such that (without loss of generality) for any element $s \in \mathcal{X}$, $f(s) \neq 0$. The curvature measures how far f is from submodularity or linearity. Define *curvature* of f over the matroid $\mathcal{I}$ as:

$$\nu_f(\mathcal{I}) \triangleq 1 - \min_{s \in \mathcal{S}, \mathcal{S} \in \mathcal{I}} \frac{f(\mathcal{S}) - f(\mathcal{S} \setminus \{s\})}{f(s)}. \tag{2}$$

Note that the definition of curvature $\nu_f(\mathcal{I})$ (Eq. 2) implies that $0 \leq \nu_f(\mathcal{I}) \leq 1$. Specifically, if $\nu_f(\mathcal{I}) = 0$, it means for all the feasible sets $\mathcal{S} \in \mathcal{X}$, $f(\mathcal{S}) = \sum_{s \in \mathcal{S}} f(s)$. In this case, f is a modular function. In contrast, if $\nu_f(\mathcal{I}) = 1$, then there exist a feasible $\mathcal{S} \in \mathcal{I}$ and an element $s \in \mathcal{X}$ such that $f(\mathcal{S}) = f(\mathcal{S} \setminus \{s\})$. In this case, the element s is redundant for the contribution of the value of f given the set $\mathcal{S} \setminus \{s\}$.

Global oblivious set and subgraph $\mathcal{G}_2$: Similarly to the construction of the *local oblivious set* on each clique in Section III, we define the *global oblivious set*, $\mathcal{S}^o$ on graph $\mathcal{G}$ as follows. $\mathcal{S}^o$ satisfies the attack cardinality constraint, i.e., $|\mathcal{S}^o| = \alpha$. Also, $\mathcal{S}^o$ is constructed sequentially such that each strategy $s \in \mathcal{X}(\mathcal{R})$ added in $\mathcal{S}^o$ per round achieves the highest value of $f(s)$ among all the strategies in $\mathcal{X}(\mathcal{R})$ and follows "*one strategy per robot*" constraint.

Then we define a subgraph of $\mathcal{G}$ as $\mathcal{G}_2$. The nodes of $\mathcal{G}_2$ are the robots whose strategies are not selected in the *global oblivious set*, i.e., $\mathcal{R} \setminus \mathcal{R}(\mathcal{S}^o)$. The communication links among the robots in $\mathcal{G}_2$ are the same as that existed in graph $\mathcal{G}$. We give an example of subgraph $\mathcal{G}_2$ in Fig. 2. We assume the number of attacks $\alpha = 2$ on graph $\mathcal{G}$ (Fig. 1) and the *global oblivious set* $\mathcal{S}^o$ (with $|\mathcal{S}^o| = 2$) contains one strategy from robot 4 and one strategy from robot 5. Then the remaining robots formulate the subgraph $\mathcal{G}_2$ with the same communication links among these robots in graph $\mathcal{G}$. Evidently, $\mathcal{G}_2$ has $N - \alpha$ robots if graph $\mathcal{G}$ has N robots, and has less or equal number of cliques than $\mathcal{G}$ does.

Notably, each clique does not know either the *global oblivious set* $\mathcal{S}^o$ or subgraph $\mathcal{G}_2$, since it has no idea of $f(s)$ from other cliques.

Next, we present the performance of Algorithm 1.

Theorem 1 (Performance of Algorithm 1). *Consider Problem 1, the notation therein, the notation in Algorithm 1, and the definitions:*

- *let $f^\star$ be the optimal value to Problem 1;*

- *given a set $\mathcal{S}$ as solution to Problem 1, let $\mathcal{A}^\star(\mathcal{S})$ be a worst-case set removal from $\mathcal{S}$, that is: $\mathcal{A}^\star(\mathcal{S}) \in \arg\min_{\mathcal{A}\subseteq\mathcal{S},|\mathcal{A}(\mathcal{S})|\leq\alpha} f(\mathcal{S}\setminus\mathcal{A})$. Evidently, a removal from $\mathcal{S}$ corresponds to a set of robot/sensor attacks;*

The performance of Algorithm 1 is bounded as follows:

1) (Approximation performance) *Algorithm 1 returns a strategy set $\mathcal{S}$ such that each robot selects a decision strategy (partition matroid constraint $\mathcal{I}$), and If $\mathcal{K}(\mathcal{G}) = 1$,*

$$\frac{f(\mathcal{S}\setminus\mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \frac{1}{2}\max[1-\nu_f(\mathcal{I}), \frac{1}{(\alpha+1)}, \frac{1}{(N-\alpha)}] \tag{3}$$

Else, $\mathcal{K}(\mathcal{G}) \geq 2$,

$$\frac{f(\mathcal{S}\setminus\mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \max[\frac{1-\nu_f(\mathcal{I})}{2}, \frac{1}{(\alpha+1)\mathcal{K}(\mathcal{G}_2)\omega(\mathcal{G}_2)}, \frac{1}{(N-\alpha)\mathcal{K}(\mathcal{G}_2)\omega(\mathcal{G}_2)}] \tag{4}$$

where $\mathcal{K}(\mathcal{G})$ and $\mathcal{K}(\mathcal{G}_2)$ are the number of non-overlapping cliques in graph $\mathcal{G}$ and its subgraph $\mathcal{G}_2$, respectively. $\omega(\mathcal{G}_2)$ is the clique number of subgraph $\mathcal{G}_2$.

2) (Running time) *Algorithm 1 runs in $O(\omega^2(\mathcal{G})D^2)$ time. $\omega(\mathcal{G})$ is the clique number of graph $\mathcal{G}$ and D is the number of candidate strategies for each robot.*

The proof of Theorem 1 is the appendix.

Approximation performance. The approximation ratio in Theorem 1 implies Algorithm 1 has the same approximation performance as the centralized submodular maximization algorithm [10, Algorithm 1] when the graph $\mathcal{G}$ only has one clique. This is because, in this extreme case, the distributed communication turns out to be a centralized communication if all robots communicate within a single clique. When graph $\mathcal{G}$ has more than one clique, the approximation ratio of Algorithm 1 depends on the number of non-overlapping cliques and the clique number of its subgraph $\mathcal{G}_2$.

Running time. Theorem 1 implies that the running time of Algorithm 1 is quadratic in the clique number of graph $\mathcal{G}$ and the number of robot's candidate strategies. Notably, the centralized resilient algorithm runs in $O(N^2D^2)$ time [10]. We know the clique number of graph $\mathcal{G}$ is less than the total number of robots N when the graph has more than one clique (not in the extreme case). Thus, Algorithm 1 runs faster than the centralized resilient algorithm as long as $\mathcal{K}(\mathcal{G}) \neq 1$.

V. Simulation

We verify the performances of the proposed algorithms by a multi-robot target tracking scenario as presented in [14], [10] where each robot must choose one trajectory from its candidate trajectory set to track targets. We present both Matlab and Gazebo evaluations of our algorithm that demonstrate the performance and the strength of our approach. Our Matlab and Gazebo implementations are available online[2].

[2] https://github.com/raaslab/distributed_resilient_target_tracking.git

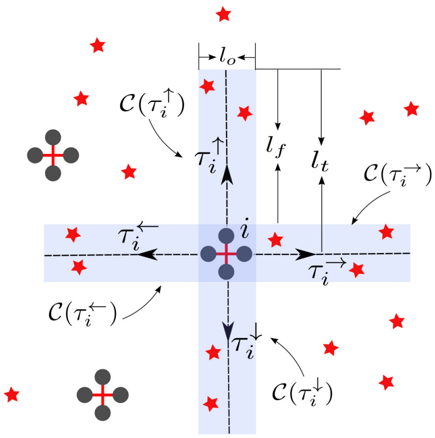


Fig. 3. Matlab simulation setup: Each robot $i \in \mathcal{R}$ (quadrotor model) has 4 possible trajectories (forward, backward, left, and right). The tracking region of each trajectory is rectangular and has the same dimensions across all 4 trajectories. We denote the tracking regions for the forward, backward, left, and right trajectories as $\mathcal{C}(\tau_i^\uparrow)$, $\mathcal{C}(\tau_i^\downarrow)$, $\mathcal{C}(\tau_i^\leftarrow)$ and $\mathcal{C}(\tau_i^\rightarrow)$ respectively; in particular, the lengths l_t and l_o denote the dimension of each rectangular tracking region; and l_f denotes the fly length for the robot. We set $l_t = l_f + l_o$ as the robot's tracking length. The red pentagrams indicate the targets.

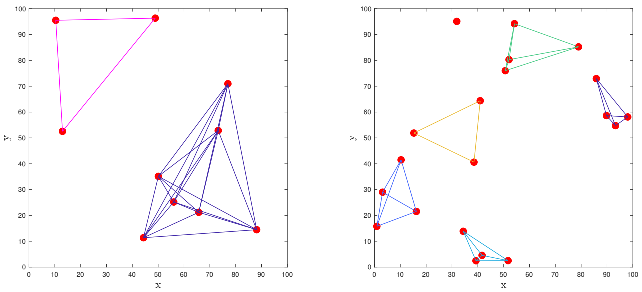
Compared algorithms. We compare our distributed resilient algorithm (Alg. 1) with two other algorithms. The algorithms differ in the communication protocol and how robots make decisions. The first algorithm is a centralized resilient algorithm, proposed in our recent work [10] where a central server communicates with all the robots and considers the worst-case robotic/sensor attack. Evidently, this algorithm is a baseline algorithm. The second algorithm is the centralized greedy algorithm. It has the same communication manner as the centralized resilient algorithm, that is, all robots can share strategies with a central server. The central server makes decisions greedily for the robots and ignores the robotic/sensor attack as proposed in [14].

Performance Sketch. The evaluations and comparisons demonstrate: (i) Algorithm 1 performs close to the centralized resilient algorithm [10]. (ii), Algorithm 1 is superior to the centralized greedy algorithm. (iii), Algorithm 1 takes less running time than the centralized resilient algorithm and the centralized greedy algorithm.

A. Matlab evaluation over one time step with static targets

We study the effect of the number of robots and of the communication range by running the algorithms over random instances of Problem 1 for a single round (one step time horizon).

Simulation setup. We consider 60 targets and a number of robots varying from 10 to 20. We set the number of attacks as 3, 4 and 8. A top view of robots and targets is shown in Fig. 3. We assume each robot moves on a fixed plane and has four candidate trajectories (strategies): forward, backward, left and right. Each robot has a square filed-of-view $l_o \times l_o$. Once a robot picks a trajectory, it flies a distance l_f along that trajectory. Thus, each trajectory has a rectangular tracking region with length $l_t = l_f + l_o$ and



(a) 10 robots with $r_c = 70$ (b) 20 robots with $r_c = 30$

Fig. 4. Matlab evaluation of distributed non-overlapping clique formulation (Algorithm 2) with different number of robots and different communication ranges. The red dot indicates the robot. The line between two robots indicates a communication link.

width l_o. We set $l_t = 10$ and $l_o = 3$ for all the robots. We randomly generate the positions of the robots and the targets in the 2D space $[0, 100] \times [0, 100] \in \mathbb{R}^2$ across 30 trials for each number of robots from 10 to 20. We assume robots have available estimates of targets' positions. For each trial, all three algorithms, Algorithm 1, centralized resilient algorithm [10] and centralized greedy algorithm [14], are executed with the same initialization, i.e., the same positions of robots and targets. Notably, for the centralized resilient algorithm and the centralized greedy algorithm, a central server chooses trajectories for all the robots. For Algorithm 1, it starts with four different communication ranges, $r_c = 30, 50, 70$ and 90. We assume each robot identifies its unique clique by the proposed clique cover approach or other clique cover algorithms based on the communication range. Then, each robot chooses one of its candidate trajectories by communicating with other robots in the same clique. All algorithms are performed for one round in each trial.

We first arbitrarily pick two trials with a varying number of robots and varying communication ranges to visualize the performance of the proposed clique cover approach in Figure 4. In these two trials, robots formulate non-overlapping cliques by communicating with neighbors only.

We examine the performance of three algorithms by the remaining tracked targets after the worst-case attack and the running time with a varying number of robots, a varying number of attacks and varying communication ranges in Fig. 5 Fig. 6. We also show the number of cliques $\mathcal{K}(\mathcal{G})$ and the clique number $\omega(\mathcal{G})$ to get a sense of the property of graph $\mathcal{G}$ in Fig. 7.

Results. The comparisons are reported in Fig. 5 and Fig. 6 with the number of attacks, $\alpha = 3, 4$ and 8 and with $r_c = 30, 50, 70$ and 90. The following observations from Fig. 5 and Fig. 6 are due:

a) Close-to-centralized communication of Algorithm 1: Fig. 5 shows that the number of targets tracked by Algorithm 1 is close to that of the centralized resilient algorithm. In particular, when the communication range is 90, Algorithm 1 works comparatively to the centralized resilient algorithm (Fig. 5-(d)). This is because, when the communication range is large, the communication graph $\mathcal{G}$ becomes dense, as shown in Fig 7-(d) where graph $\mathcal{G}$ has around 2 cliques and its clique number is over 8. Thus, Algorithm 1 performs comparatively to the centralized resilient algorithm.

b) Superior-to-centralized greedy algorithm: Fig. 5-(b), (c) & (d) shows Algorithm 1 performs better than the centralized greedy algorithm in terms of the number of targets tracked. Even when the communication range is 30 (the graph $\mathcal{G}$ is sparse with more cliques and a smaller clique number as shown in Fig 7-(a)), Algorithm 1 outperforms the centralized greedy algorithm when the number of robots is larger than 13 (Fig. 5-(a)).

c) Superior-to-centralized algorithms in the computational time: Fig. 6 shows Algorithm 1 runs faster than two centralized communication algorithms. Because, in Algorithm 1, robots select trajectories within each clique in parallel. While in the centralized communication algorithms, a central server evaluates the strategy sets from all robots in the network and schedules trajectories for them, and thus takes more time.

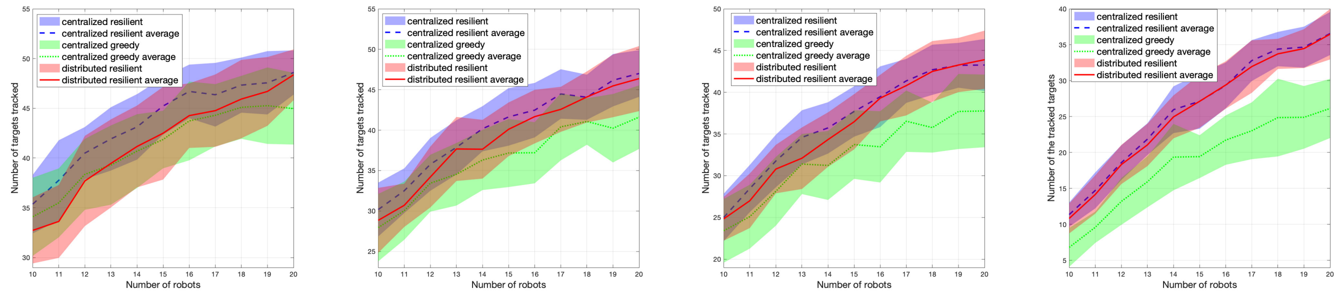
B. Gazebo evaluation over multiple steps with mobile targets

We verify the performance of Algorithm 1 by running the algorithms across multiple time steps. We consider the kinematics and dynamics of the robots, the sensing noise, and the actual trajectories of the targets.

Simulation setup. We consider a scenario where 10 aerial robots are tasked to track 50 ground mobile targets (Fig. 8-(a)). We set the number of attacks α equal to 4 and set the communication range for robots as $r_c = 5$ units in the gazebo environment. Notably, we only consider 2D $[x, y]$ coordinates for robots to identify neighbors by using the communication range. We also visualize the robots, their field-of-view, the cliques, and the targets using the Rviz environment (Fig. 8-(b)): in particular, we visualize the robots as spherical markers, their field-of-views as colored squares. For a group of robots belonging to the same clique, we set the same color for them, so as their field-of-view. Similarly to the Matlab simulation setting, each robot has 4 trajectories (forward, backward, left, and right), and flies on a different fixed plane (to avoid collision with other robots). Moreover, we set the tracking length $l_t = 6$ and tracking width $l_o = 3$ for all robots. We assume robots can obtain noisy position measurements of the targets, and then use a Kalman filter for estimate updating.

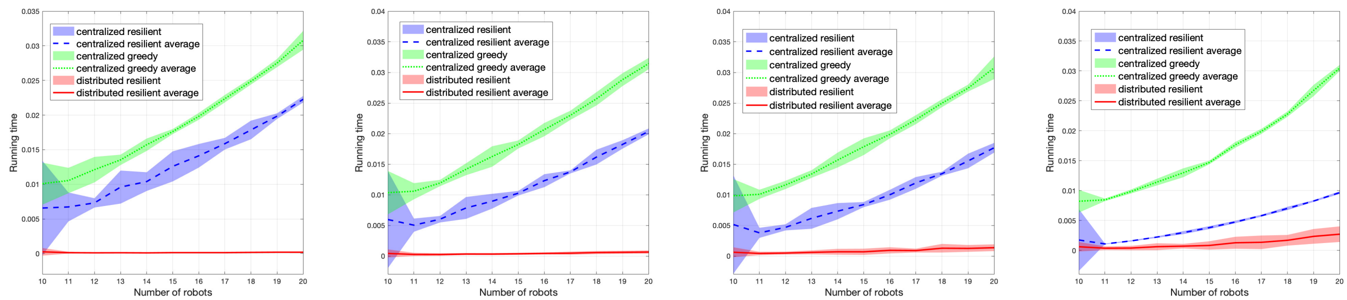
We compare the performance of Algorithm 1 with the centralized resilient algorithm [10] and the centralized greedy algorithm [14]. For each algorithm, at each time step, each robot picks one of its 4 candidate trajectories. Then all robots fly a $l_f = 3$ distance along the selected trajectory. If an attack happens, we assume the attacked robot's tracking sensor (e.g, camera) is blocked; nevertheless, we assume that it can be active again at the next time step, so that at each round the worst-case set of α robots is considered attacked. We repeat this process for 50-time steps.

We capture the performance of each algorithm with the expected number of targets tracked and the computational time for all time steps. We compare the algorithms with



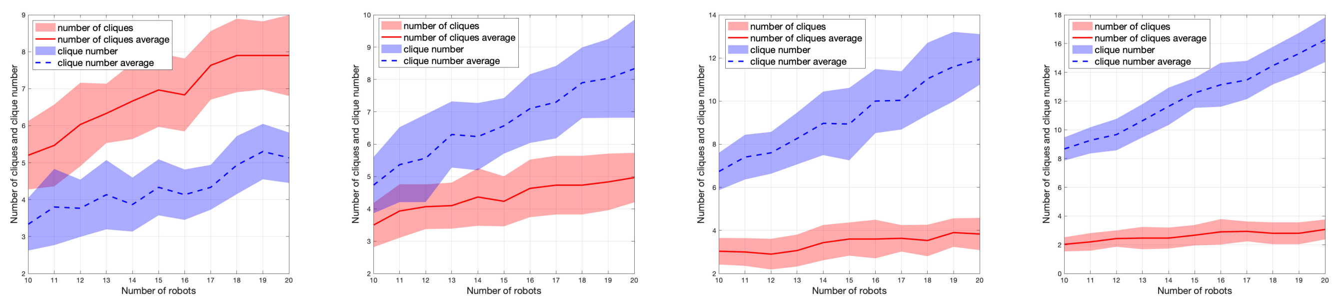
(a) $\alpha = 3$ and $r_c = 30$ (b) $\alpha = 4$ and $r_c = 50$ (c) $\alpha = 4$ and $r_c = 70$ (d) $\alpha = 8$ and $r_c = 90$

Fig. 5. Matlab evaluations: Permanence comparison of Algorithm 1 with the centralized resilient algorithm and the centralized greedy algorithm by the number of tracked targets with three different number of attacks, $\alpha = 3, 4$ and 8, and four different communication ranges $r_c = 30, 50, 70$ and 90.



(a) $\alpha = 3$ and $r_c = 30$ (b) $\alpha = 4$ and $r_c = 50$ (c) $\alpha = 4$ and $r_c = 70$ (d) $\alpha = 8$ and $r_c = 90$

Fig. 6. Matlab evaluations: Permanence comparison of Algorithm 1 with the centralized resilient algorithm and the centralized greedy algorithm by the running time with three different number of attacks, $\alpha = 3, 4$ and 8, and four different communication ranges $r_c = 30, 50, 70$ and 90.



(a) $r_c = 30$ (b) $r_c = 50$ (c) $r_c = 70$ (d) $r_c = 90$

Fig. 7. Matlab evaluations: Visualization of the number of cliques and the clique number on graph $\mathcal{G}$ with four different communication ranges $r_c = 30, 50, 70$ and 90.

respect to the average and the standard deviation of these two performance indexes. A video for this implementation is available online[3].

Results. The comparison results are reported in Fig. 9. The following observations from Fig. 9 are due:

a) Close-to-centralized communication of Algorithm 1 and better than the centralized greedy algorithm: Fig. 9-(a) shows that the number of targets tracked by Algorithm 1 is close to that of the centralized resilient algorithm, and is larger than that of the centralized greedy algorithm when the number of cliques and clique number of graph $\mathcal{G}$ are 3 and 5 on average (Fig. 9-(c)).

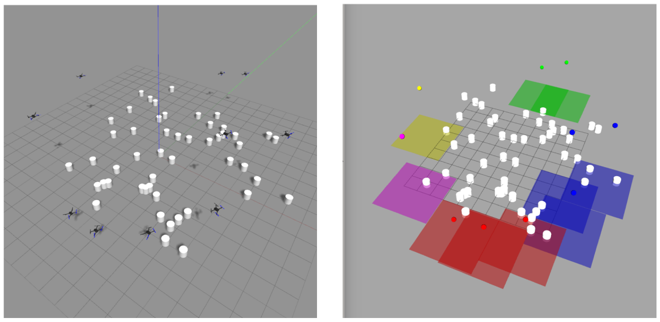
b) Superior-to-centralized algorithms in the running time: Fig. 9-(b) shows Algorithm 1 runs faster than two centralized communication algorithms when the number of cliques and clique number of graph $\mathcal{G}$ are 3 and 5 on average (Fig. 9-(c)).

All in all, in the above simulations, Algorithm 1 achieves a close-to-centralized communication performance and runs faster.

VI. Conclusion

We studied a submodular maximization problem where a group of decision makers with a limited communication ability, collaboratively select strategies to maximize a common

[3]`https://youtu.be/cGCUErXESZ0`



(a) Gazebo environment (b) Rviz environment

Fig. 8. Gazebo simulation setup: 10 aerial robots and 50 ground mobile targets: (a) Gazebo environment; and (b) Rviz environment, where: each aerial robot is color-coded, and its coverage region is depicted with the same color. The robots in the same clique are depicted with the same color. The communication range r_c is set as 5 units in the gazebo environment. The targets are depicted as white cylindrical markers.

objective function against a known number of the worst-case attacks. We proposed a distributed resilient algorithm that has a provable performance guarantee and runs efficiently in polynomial time for such problem. We demonstrated the performance of our algorithm by implementing a multi-robot target tracking scenario in both Matlab and Gazebo simulations. Notably, the results of this paper can be extended to any other submodular maximization applications involved with a group of decision makers with limited information to play against attacks.

By confining the communication within each unique clique, we sometimes weaken the communication ability of the robot, since some robot may have neighbors in other cliques whom it can communicate with. Thus, our first ongoing work is to improve the proposed distributed approach by making the robot select strategy based on the decisions from all its neighbors, similarly to the distributed greedy algorithm in [19], [20] but not in a sequential manner. Our second line of future work focuses on an unknown number of attacks, e.g., captured by a stochastic process [28].

References

[1] M. Govindarasu, A. Hann, and P. Sauer, "Cyber-physical systems security for smart grid," *Future Grid Initiative White Paper, PSERC, Feb*, 2012.

[2] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[3] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1039–1046, 2017.

[4] C. Symantec. (2014) The cyber resilience blueprint: A new perspective on security. [Online]. Available: https://www.symantec.com/content/en/us/enterprise/white_papers/b-cyber-resilience-blueprint-wp-0814.pdf

[5] S. IBM. (2018) Cyber resilience services. [Online]. Available: https://www.ibm.com/services/business-continuity/cyber-resilience

[6] C. MITRE. Cybersecurity: Resiliency. [Online]. Available: https://www.mitre.org/capabilities/cybersecurity/resiliency

[7] Q. Zhu and T. Başar, "Robust and resilient control design for cyber-physical systems with an application to power systems," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 4066–4071.

[8] H. Alemzadeh, C. Di Martino, Z. Jin, Z. T. Kalbarczyk, and R. K. Iyer, "Towards resiliency in embedded medical monitoring devices," in *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*. IEEE, 2012, pp. 1–6.

[9] Y. Yuan, Q. Zhu, F. Sun, Q. Wang, and T. Başar, "Resilient control of cyber-physical systems against denial-of-service attacks," in *2013 6th International Symposium on Resilient Control Systems (ISRCS)*. IEEE, 2013, pp. 54–59.

[10] 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, 2019.

[11] D. Saldana, A. Prorok, S. Sundaram, M. F. Campos, and V. Kumar, "Resilient consensus for time-varying networks of dynamic agents," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 252–258.

[12] L. Guerrero-Bonilla, D. Saldana, and V. Kumar, "Design guarantees for resilient robot formations on lattices," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 89–96, 2019.

[13] 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.

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

[15] H. Ding and D. Castanón, "Multi-agent discrete search with limited visibility," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 108–113.

[16] 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.

[17] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functionsii," in *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.

[18] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *computer*, vol. 35, no. 10, pp. 54–62, 2002.

[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, 2018.

[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] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Resilient non-submodular maximization over matroid constraints," *arXiv preprint arXiv:1804.01013*, 2018.

[22] V. Tzoumas, K. Gatsis, A. Jadbabaie, and G. J. Pappas, "Resilient monotone submodular function maximization," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1362–1367.

[23] 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.

[24] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," in *Advances in Neural Information Processing Systems*, 2013, pp. 2049–2057.

[25] R. B. Myerson, *Game theory*. Harvard university press, 2013.

[26] F. V. Fomin and D. Kratsch, *Exact exponential algorithms*. Springer Science & Business Media, 2010.

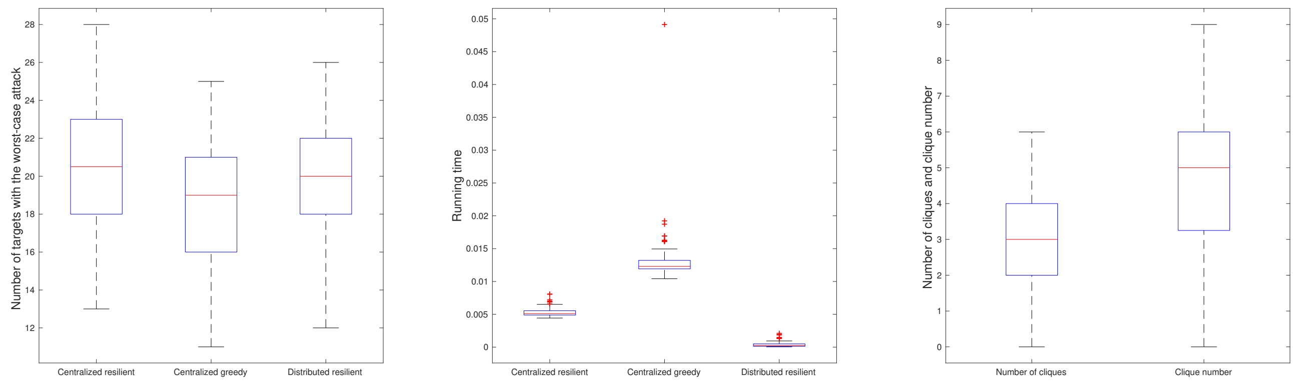
[27] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem," *Discrete applied mathematics*, vol. 7, no. 3, pp. 251–274, 1984.

[28] H. Park and S. Hutchinson, "Robust rendezvous for multi-robot system with random node failures: an optimization approach," *Autonomous Robots*, pp. 1–12, 2018.

[29] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," *Mathematical Programming*, vol. 172, no. 1-2, pp. 505–537, 2018.

Appendix

*a) **Distributed clique cover**:* We execute our distributed non-overlapping clique cover algorithm in three commu-



(a) Number of tracked targets (b) Running time (c) Number of cliques & clique number

Fig. 9. Gazebo evaluations: Comparison (average and standard deviation across the 50 rounds) of Algorithm 1 with the centralized resilient algorithm and the centralized greedy algorithm. Performance is captured by the expected number of tracked targets. Fig. 9-(b) compares the running time of three algorithms. Fig. 9-(c) illustrates the number of cliques is around 3 and the clique number is around 5 when the communication range $r_c = 5$ units.

nication rounds and one computation round as shown in Algorithm 2.

1) *1st communication*: each robot i finds its neighbors within its communication range as $\mathcal{N}_i$ (Alg. 2, line 2). It then stores its neighbors and itself in the set $\mathcal{N}_i^+$ (Alg. 2, line 3).
2) *2nd communication*: each robot i shares the set $\mathcal{N}_i^+$ with all of its neighbors (Alg. 2, line 4). After the sharing, it receives all $\mathcal{N}_j^+, j \in \mathcal{N}_i$ from its neighbors (Alg. 2, line 5). We arbitrarily order these sets as $\mathcal{N}_{j1}^+, \cdots, \mathcal{N}_{j|\mathcal{N}_i|}^+$ for the convenience of expression. It then stores the set $\mathcal{N}_i^+$ and all $\mathcal{N}_j^+$ from its neighbors in a superset $\mathcal{N}^+ = \{\mathcal{N}_i^+, \mathcal{N}_{j1}^+, \cdots, \mathcal{N}_{j|\mathcal{N}_i|}^+\}$ (Alg. 2, line 6).
3) *computation*: Given the superset $\mathcal{N}^+$, each robot i first finds all of its maximal cliques by computing the intersections among m subsets in $\mathcal{N}^+$ from $m = |\mathcal{N}^+|$ to $m = 2$ in a loop (Alg. 2, lines 8-20). With a slight abuse of notation, we denote $|\mathcal{N}^+|$ as the number of the subsets in $\mathcal{N}^+$. In the loop, for each value of m, there are $\binom{|\mathcal{N}^+|}{m}$ possible intersections (Alg. 2, line 9). The robot i stores all the intersections whose cardinality equals to m in a superset $\mathcal{C}^i$ (Alg. 2, lines 10-15). Once the superset $\mathcal{C}^i$ is non-empty, the robot i sets its maximal clique set $\mathcal{C}^{i*}$ as $\mathcal{C}^i$ (Alg. 2, lines 16-17). The loop with descending order of m terminates (Alg. 2, line 18). After finding all maximal cliques, the robot i decides its unique maximal clique. If there is only one intersection (subset) in $\mathcal{C}^i$, the robot i sets its unique maximal clique $\mathcal{C}^{iu}$ as $\mathcal{C}^{i*}$ (Alg. 2, lines 21-22). Otherwise, it computes the neighbors for all the intersections (subsets) in $\mathcal{C}^{i*}$. Here, we compute the neighbors of a set of robots as the union of the neighbors from all the individual robots within this set. It then picks the intersection which has fewest neighbors as its unique maximal clique $\mathcal{C}^{iu}$ (Alg. 2, line 24).
4) *3rd communication*: each robot i shares its unique maximal clique with all of its neighbors (Alg. 2, line 26).

The robots with the same unique maximal clique formulate a unique clique on graph $\mathcal{G}$. A unique clique can have one robot only. After all unique cliques are identified, the non-overlapping clique cover is achieved on the graph $\mathcal{G}$. Since each robot finds its unique clique based on the local information only, Algorithm 2 is a feasible or sub-optimal (not global optimal) solution for covering the graph with the minimum number of non-overlapping cliques. However, the strength of Algorithm 2 is the way that the robot decides its unique maximal clique when it has two or more maximal cliques. Since the robot picks the maximal clique with fewest neighbors, it increases the potential for other maximal cliques to formulate other larger cliques, which leads to a higher chance of generating fewer and larger non-overlapping cliques.

*b) **Proof for Theorem 1**:* We first provide the following notations for the convenience of the proof.

Denote the optimal selection as $\mathcal{S}^\star$ with $|\mathcal{S}^\star| \leq N$. $\mathcal{S}^\star = \bigcup_{k=1}^{\mathcal{K}} \mathcal{S}_k^\star$ where $\mathcal{S}_k^\star,\ k \in \{1, \cdots, \mathcal{K}\}$ is the set selected by the optimal solution in each clique $\mathcal{C}_k$, and $|\mathcal{S}_k^\star| \leq n_k$. Denote the worst-case attack on the optimal set $\mathcal{S}^\star$ with respect to the whole network $\mathcal{G}$ as $\mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})$ with $|\mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})| \leq \alpha$. Similarly, denote the set selected by the Algorithm 1 as $\mathcal{S}$ with $|\mathcal{S}| \leq N$. $\mathcal{S} = \bigcup_{k=1}^{\mathcal{K}} \mathcal{S}_k$ where $\mathcal{S}_k,\ k \in \{1, \cdots, \mathcal{K}\}$ is the set selected by each clique $\mathcal{C}_k$ with $|\mathcal{S}_k| \leq n_k$. Denote the worst-case attack on the chosen set $\mathcal{S}$ with respect to the overall graph as $\mathcal{A}^\star(\mathcal{S}|\mathcal{G})$ with $|\mathcal{A}^\star(\mathcal{S}|\mathcal{G})| \leq \alpha$.

Proof of approximation ratio. In Algorithm 1, all cliques of robots select out strategy set $\mathcal{S}$ in parallel. In fact, these cliques together rank out the *global oblivious set*. That is because, the oblivious decision from all cliques are the union set of all *local oblivious sets*, which is a superset of the *global oblivious set*. But this *global oblivious set* is unknown to each

Algorithm 2: Distributed Non-overlapping Clique Cover

Input:
- set of robots $\mathcal{R}$
- positions the robots
- communication range r_c

Output: Non-overlapping clique cover on graph $\mathcal{G}$

for each robot i **do**
  finds its neighbor set $\mathcal{N}_i$ within r_c
  sets $\mathcal{N}_i^+ = \{i, \mathcal{N}_i\}$
  shares $\mathcal{N}_i^+$ with its neighbors
  receives all $\mathcal{N}_j^+, j \in \mathcal{N}_i$
  stores $\mathcal{N}_i^+$ and all $\mathcal{N}_j^+(s)$ in $\mathcal{N}^+ = \{\mathcal{N}_i^+, \mathcal{N}_{j1}^+, \cdots, \mathcal{N}_{j|\mathcal{N}_i|}^+\}$
  $\mathcal{C}^i \leftarrow \emptyset$;
  for $m = |\mathcal{N}^+| : 2$
    compute all combinations of m subsets from $\mathcal{N}^+$ as $\binom{\mathcal{N}^+}{m}$
    for each combination in $\binom{\mathcal{N}^+}{m}$ **do**
      computes its intersection, $\mathcal{C}_{\text{intersect}}$
      if $|\mathcal{C}_{\text{intersect}}| = m$ **do**
        puts $\mathcal{C}_{\text{intersect}}$ in $\mathcal{C}^i$
      end if
    end for
    if $\mathcal{C}^i$ is not empty
      sets its maximal clique set $\mathcal{C}^{i*} = \mathcal{C}^i$
      breaks the **for loop** of m
    end if
  end for
  if $\mathcal{C}^{i*}$ only has one subset **do**
    sets its unique maximal clique as $\mathcal{C}^{iu} = \mathcal{C}^{i*}$
  else
    chooses the subset of $\mathcal{C}^{i*}$ that has fewest neighbors
  end if
  shares its unique clique with all of its neighbors
end for

clique. We partition the strategy set $\mathcal{S}$ from Algorithm 1 into the *global oblivious set* $\mathcal{S}^o$ and $\mathcal{S}_2$. $\mathcal{S}_2 = \mathcal{S} \setminus \mathcal{S}^o$, corresponding to the strategies on the subgraph $\mathcal{G}_2$.

We prove the approximation ratio of Algorithm 1 by proving the following three inequalities.

$$\frac{f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \frac{1-\nu_f(\mathcal{I})}{2} f(\mathcal{S}^\star \setminus \mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})), \tag{5}$$

$$\text{If } \mathcal{K}(\mathcal{G}) = 1 : \frac{f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \frac{1}{2}\max[\frac{1}{\alpha+1}, \frac{1}{N-\alpha}] f(\mathcal{S}^\star \setminus \mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})), \tag{6}$$

$$\text{If } \mathcal{K}(\mathcal{G}) \geq 2 : \frac{f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \max[\frac{1}{\alpha+1}, \frac{1}{N-\alpha}] \frac{1}{\mathcal{K}(\mathcal{G}_2)} \frac{1}{\omega(\mathcal{G}_2)} f(\mathcal{S}^\star \setminus \mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})). \tag{7}$$

Note that the number of cliques in subgraph $\mathcal{G}_2$, $\mathcal{K}(\mathcal{G}_2) \geq 1$, since $\alpha < N$. We start with the proof of the ineq. 5 by using the property of the curvature $\nu_f(\mathcal{I})$.

$$\begin{aligned}
& f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}|\mathcal{G})) \\
& \geq (1-\nu_f(\mathcal{I})) \sum_{a \in \mathcal{S}_2|\mathcal{G}_2} f(a) && (8) \\
& \geq (1-\nu_f(\mathcal{I})) \sum_{a \in \mathcal{S}_2^g|\mathcal{G}_2} f(a) && (9) \\
& \geq (1-\nu_f(\mathcal{I})) f(\mathcal{S}_2^g|\mathcal{G}_2) && (10) \\
& \geq \frac{1-\nu_f(\mathcal{I})}{2} f(\mathcal{S}_2^\star|\mathcal{G}_2) && (11) \\
& \geq \frac{1-\nu_f(\mathcal{I})}{2} f(\mathcal{S}^\star \setminus \mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})) && (12)
\end{aligned}$$

where eqs. 8 - 12 hold for the following reasons. Ineq. 8 follows from [21, Lemma 2 and the proof of Theorem 1]. It is based on the property of the curvature and the fact that every element in $\mathcal{S}^o$ is larger than every element in $\mathcal{S}_2|\mathcal{G}_2$. In ineq. 9, $\mathcal{S}_2^g|\mathcal{G}_2$ denotes the strategy set selected by the greedy algorithm with centralized communication on the subgraph $\mathcal{G}_2$. While $\mathcal{S}_2|\mathcal{G}_2$ is the strategy set selected by the resilient algorithm (Alg. 1) on the subgraph $\mathcal{G}_2$. We know the greedy algorithm with centralized communication avoids more redundancy than the distributed resilient algorithm (Alg. 1) does. Thus, the value $f(a), a \in \mathcal{S}_2$ is always larger or equal to the value $f(a'), a' \in \mathcal{S}_2^g$ where a and a' are the strategies from the same robot. Then, ineq. 9 holds. Ineq. 10 holds from the submodularity of the function f. Ineq. 11 holds from the property of the greedy algorithm [27, Theorem 2.3] where $\mathcal{S}_2^\star|\mathcal{G}_2$ denotes the optimal strategy set on the subgraph $\mathcal{G}_2$. Finally, ineq. 12 holds from [29, Lemma 2].

When the communication graph $\mathcal{G}$ only has one clique, Algorithm 1 is exactly the same as the centralized resilient algorithm from [10, Algorithm 1], and thus ineq. 6 holds accordingly from [10, Theorem 1].

Finally, we prove the third ineq. 7 as follows.

$$\begin{aligned}
& f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}|\mathcal{G})) \\
& \geq \gamma f(\mathcal{S}_2|\mathcal{G}_2) && (13) \\
& \geq \gamma \frac{1}{\mathcal{K}(\mathcal{G}_2)} \sum_{k=1}^{\mathcal{K}(\mathcal{G}_2)} f(\mathcal{S}_{2,k}|\mathcal{C}_k(\mathcal{G}_2)) && (14) \\
& \geq \gamma \frac{1}{\mathcal{K}(\mathcal{G}_2)} \sum_{k=1}^{\mathcal{K}(\mathcal{G}_2)} h_k f(\mathcal{S}_{2,k}^\star|\mathcal{C}(\mathcal{G}_2)) && (15) \\
& \geq \gamma \frac{1}{\mathcal{K}(\mathcal{G}_2)} \min_k h_k \sum_{k=1}^{\mathcal{K}(\mathcal{G}_2)} f(\mathcal{S}_{2,k}^\star|\mathcal{C}_k(\mathcal{G}_2)) && (16) \\
& \geq \gamma \frac{1}{\mathcal{K}(\mathcal{G}_2)} \frac{1}{\omega(\mathcal{G}_2)} f(\mathcal{S}_2^\star|\mathcal{G}_2) && (17) \\
& \geq \gamma \frac{1}{\mathcal{K}(\mathcal{G}_2)} \frac{1}{\omega(\mathcal{G}_2)} f(\mathcal{S}^\star \setminus \mathcal{A}^\star(\mathcal{S}^\star|\mathcal{G})) && (18)
\end{aligned}$$

where $k \in \{1, \cdots, \mathcal{K}(\mathcal{G}_2)\}$, $\gamma = \max[\frac{1}{\alpha+1}, \frac{1}{N-\alpha}]$ and

$$h_k = \begin{cases} \frac{1}{n_{2,k}}, & \text{if } n_{2,k} = 1 \text{ or } \\ & \mathcal{C}_k(\mathcal{G}_2) \text{ has an oblivious decision,} \\ \frac{1}{2}, & \text{else.} \end{cases}$$

$\mathcal{S}_{2,k}|\mathcal{C}_k(\mathcal{G}_2)$ is the strategy set by Algorithm 1 on the clique $\mathcal{C}_k(\mathcal{G}_2)$ and $n_{2,k}$ is its carnality. Eqs. 13-18 hold for the following reasons. Ineq. 13 holds from [21, the proof of Theorem 1]. Ineq. 14 holds from the monotonicity of the submodular function f: $f(\mathcal{S}_2|\mathcal{G}_2) \geq f(\mathcal{S}_{2,k}|\mathcal{C}_k(\mathcal{G}_2))$ for all $k \in \{1, \cdots, \mathcal{K}(\mathcal{G}_2)\}$. Ineq. 15 holds from the distributed resilient algorithm (Alg. 1) in each clique. As long as the clique $\mathcal{C}_k(\mathcal{G}_2)$ has an oblivious strategy, $f(\mathcal{S}_{2,k}|\mathcal{C}_k(\mathcal{G}_2)) \geq \frac{1}{n_{2,k}} f(\mathcal{S}^\star_{2,k}|\mathcal{C}(\mathcal{G}_2))$. That is because, the oblivious decision picks the strategy with the largest contribution without considering the redundancy. There is a special case where the *local oblivious set* of the clique $\mathcal{C}_k(\mathcal{G})$ is exactly the *global oblivious set*. In this case, the clique $\mathcal{C}_k(\mathcal{G}_2)$ only performs a greedy algorithm. Because the *local oblivious set* of $\mathcal{C}_k(\mathcal{G})$ is picked as the *global oblivious set* and the remaining robots, $\mathcal{C}_k(\mathcal{G}_2)$ execute a greedy algorithm only. In this case, $f(\mathcal{S}_{2,k}|\mathcal{C}_k(\mathcal{G}_2)) \geq \frac{1}{2} f(\mathcal{S}^\star_{2,k}|\mathcal{C}(\mathcal{G}_2))$ which is from the property of the greedy algorithm [27, Theorem 2.3] with $\mathcal{S}^\star_{2,k}|\mathcal{C}_k(\mathcal{G}_2)$ denoting the optimal strategy set on the clique $\mathcal{C}_k(\mathcal{G}_2)$. Also, when $n_{2,k} = 1$, both an oblivious strategy and a greedy strategy are the same as the optimal strategy. Ineq. 16 holds obviously from ineq. 15. To explain ineq. 17, we compute $\min_k h_k$ as

$$\begin{cases} \frac{1}{\omega(\mathcal{G}_2)}, & \text{if } n_{2,k} = 1 \text{ for all } k \in \{1, \cdots, \mathcal{K}(\mathcal{G}_2)\} \\ & \text{or all } \mathcal{C}_k(\mathcal{G}_2)\ k \in \{1, \cdots, \mathcal{K}(\mathcal{G}_2)\} \\ & \text{have at least an oblivious decision,} \\ \min_k[\frac{1}{n_{2,k}}, \frac{1}{2}], & \text{else: there exists one clique } \mathcal{C}_k(\mathcal{G}_2) \\ & \text{that only has greedy strategy.} \end{cases}$$

Given $\omega(\mathcal{G}_2)$ is the clique number of the subgraph $\mathcal{G}_2$, we have $\omega(\mathcal{G}_2) \geq n_{2,k}$ for all k. As long as there exists one clique $\mathcal{C}_k(\mathcal{G}_2)$ containing more than one robot, $n_{2,k} \geq 2$. Thus, overall, $\min_k h_k \geq \frac{1}{\omega(\mathcal{G}_2)}$, and therefore ineq. 17 holds from the submodularity of the function f. Ineq. 18 holds from [29, Lemma 2].

Combining the proofs of ineqs. 5, 6, and 7, we prove the approximation ratio in Theorem 1.

Proof of running time. Since all cliques perform a resilient algorithm in parallel by Algorithm 1, we only focus on the clique which has the largest number of robots. We know the number of the robots in this largest clique is the clique number of the graph, $\omega(\mathcal{G})$. Given each robot has D candidate strategies, the oblivious approach takes $O(\omega(\mathcal{G})D\log(\omega(\mathcal{G})D))$ evaluations to rank out the *local oblivious set* by using quick sort. And then the greedy approach takes $O((\omega(\mathcal{G}) - \alpha)^2 D^2)$ evaluations for the remaining robots if $\omega(\mathcal{G}) > \alpha$. If $\omega(\mathcal{G}) \leq \alpha$, the greedy algorithm has nothing to do, and thus takes 0 evaluations. Thus, overall, algorithm 1 takes $O(\omega^2(\mathcal{G})D^2)$ evaluations.