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

Devising a resilient system is receiving an increasing interest in academia [1], [2], [3] as well as 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].

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]. 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 [13], and the robot has the limited sensing and communication range so that it can make decisions based on the local information only [14], [15]. 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 [16], 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 [17], [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 *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 [18], [17], [19], [10] by proposing the distributed submodular maximization; and go beyond the distributed submodular maximization [16], [14], [15] by proposing the resilient submodular maximization.

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 of the largest clique in graph G.

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.

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 [20]. 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 $\mathcal{T}$ as $\mathcal{R}(\mathcal{T})$ 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 $\mathcal{G}$. There is no communication allowed (or required) between cliques. Denote $\mathcal{S} = \bigcup_{i \in N} s_i$ as the strategy set selected by all cliques of robots. Denote $\mathcal{S}_k$ and n_k as the strategy set and the number of robots in each clique $k, k \in \{1, \cdots, \mathcal{K}(\mathcal{G})\}$. We assume there exists a known number of the worst-case attacks, α, to the whole robot team. If the robot i is attacked, its strategy s_i will stop contributing to the function $f(\cdot)$. Formally, an adversary attacks the robot team by removing a subset of trajectories, $\mathcal{A}$, from the trajectory set $\mathcal{S}$. After the attack, the function value on the trajectory set $\mathcal{S}$, $f(\mathcal{S})$ is reduced to $f(\mathcal{S} \setminus \mathcal{A})$. We also assume that the number of attacks, α, is less than the number of total robots, N. Each clique only knows the total number of the attacks, α, and has no idea of how α attacks are distributed among cliques. The objective is to maximize a submodular function defined on the strategy set selected by the robots against the worst-case attacks. We propose the problem as below:

Problem 1 (Distributed Resilient Submodular Maximization).

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

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 $\mathcal{X}_i$. The constraint $|\mathcal{A}| \leq \alpha$ captures the problem assumption that at most α robots in the network can fail or get attacked.

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.

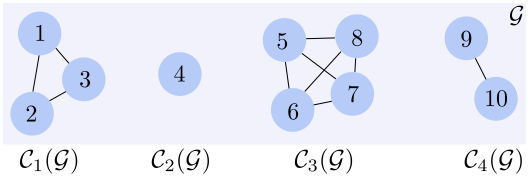


Fig. 1. A graph $\mathcal{G}$ contains 10 robots and 4 cliques.

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 full version of this paper[2]). Notably, this clique partition problem is also called "*non-overlapping clique cover*" in the literature, which is NP-hard even for a centralized solution [21]. Thus, we assume for a stationary communication graph, each robot knows its unique clique. 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 in Algorithm 1.

After the non-overlapping clique cover, all cliques of robots work in parallel to perform against the attacks (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 (line 3), it sets the number of attacks as α. A resilient algorithm is executed by a combination of an oblivious decision (lines 4-9) and a greedy decision (lines 10-15).
2) If the number of attacks is larger than the clique's size (line 16), the clique sets the number of attacks as its size. Only an oblivious decision is made (lines 17-22).

[2]A full version of this paper involving a distributed clique cover algorithm, all the proofs, and Matlab simulation is available online: https://www.raas.ece.vt.edu/wordpress/wp-content/uploads/2019/04/MRS19_full.pdf

Algorithm 1: Distributed Resilient Submodular Maximization

Input:
- set of robots $\mathcal{R}$
- robot strategy 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: $\mathcal{S} = \bigcup_{k=1}^{\mathcal{K}(\mathcal{G})} \mathcal{S}_k$

3) The strategy set in this clique is the union set of the *local oblivious set* and the *local greedy set* (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 (line 27).

IV. Performance analysis

We quantify the performance of Algorithm 1, by bounding its approximation ratio and the running time.

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 strategy (partition matroid constraint $\mathcal{I}$), and*
If $\mathcal{K}(\mathcal{G}) = 1$,

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

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

$$\frac{f(\mathcal{S} \setminus \mathcal{A}^\star(\mathcal{S}))}{f^\star} \geq \max[\frac{1 - \nu_f(\mathcal{I})}{2}, \frac{1}{(\alpha+1)\mathcal{K}(\mathcal{G}_2)\omega(\mathcal{G}_2)}, \frac{1}{(N-\alpha)\mathcal{K}(\mathcal{G}_2)\omega(\mathcal{G}_2)}] \quad (3)$$

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 $\mathcal{G}_2$ (see the definition in the full version), respectively. $\omega(\mathcal{G}_2)$ is the clique number of subgraph $\mathcal{G}_2$. $\nu_f(\mathcal{I})$ is the curvature of submodular f defined on matroid constraint $\mathcal{I}$ (see the definition in the full version).

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.*

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 $\mathcal{K}(\mathcal{G}) \neq 1$.

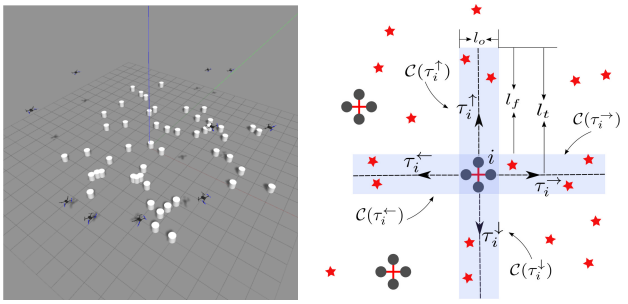
V. Simulation

We verify the performances of the proposed algorithms by a multi-robot target tracking scenario as presented in [22], [10] where each robot must choose one trajectory from its candidate trajectory set to track targets. We present both Matlab[3] and Gazebo evaluations of our algorithm that demonstrate the performance and the strength of our approach. Our Matlab and Gazebo implementations are available online[4].

Gazebo simulation. We verify the performance of Algorithm 1 by running the algorithms across multiple time steps.

[3]Due to the limited space here, we provide an extensive Matlab simulation studying the effect of the number of robots, the number of attacks, and the communication range in the full version of this paper.

[4]`https://github.com/raaslab/distributed_resilient_target_tracking.git`



(a) Gazebo environment (b) A top view

Fig. 2. Gazebo simulation setup: 10 aerial robots and 50 ground mobile targets: (a) Gazebo environment; and (b) A top view: 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.

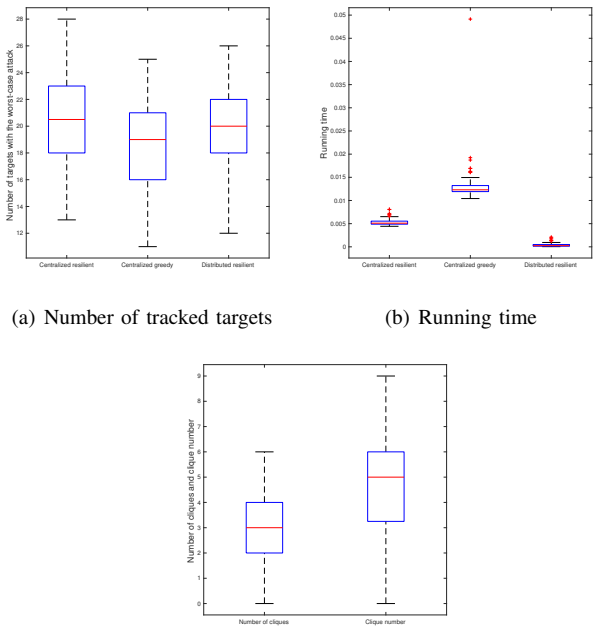
We consider the kinematics and dynamics of the robots, the sensing noise, and the actual trajectories of the targets.

We consider a scenario where 10 aerial robots are tasked to track 50 ground mobile targets (Fig. 2-(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. For a clear visualization, we draw a top view of robots and targets in Fig. 2-(b). We assume each robot has 4 trajectories (forward, backward, left, and right), and flies on a different fixed plane (to avoid collision with other robots). 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 width l_o. 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 [22]. 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 respect to the average and the standard deviation of these two performance indexes. A video for this implementation is available online[5].

[5]https://youtu.be/cGCUErXESZ0



(a) Number of tracked targets (b) Running time

(c) Number of cliques & clique number

Fig. 3. 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. 3-(b) compares the running time of three algorithms. Fig. 3-(c) illustrates the number of cliques is around 3 and the clique number is around 5 when the communication range $r_c = 5$ units.

Results. The comparison results are reported in Fig. 3. The following observations from Fig. 3 are due:

a) Close-to-centralized communication of Algorithm 1 and better than the centralized greedy algorithm: Fig. 3-(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. 3-(c)).

b) Superior-to-centralized algorithms in the running time: Fig. 3-(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. 3-(c)).

All in all, 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 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.

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

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.