2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Resilient Monotone Submodular Function Maximization

Vasileios Tzoumas,[1] Konstantinos Gatsis,[1] Ali Jadbabaie,[2] George J. Pappas[1]

***Abstract*— In this paper, we focus on applications in machine learning, optimization, and control that call for the resilient selection of a few elements, e.g. features, sensors, or leaders, against a number of adversarial denial-of-service attacks or failures. In general, such resilient optimization problems are hard, and cannot be solved exactly in polynomial time, even though they often involve objective functions that are monotone and submodular. Notwithstanding, in this paper we provide the first scalable algorithm for their approximate solution, that is valid for any number of attacks or failures, and which, for functions with low curvature, guarantees superior approximation performance. Notably, the curvature has been known to tighten approximations for several non-resilient maximization problems, yet its effect on resilient maximization had hitherto been unknown. We complement our theoretical analyses with supporting empirical evaluations.**

I. Introduction

During the last decade, researchers in machine learning, optimization, and control have focused on questions such as:

- *(Sensor selection)* How many sensors do we need to deploy in a large water distribution network to detect a contamination outbreak as fast as possible? [1];
- *(Feature selection)* Which few features do we need to select from the data flood to optimally train a spam e-mail classifier? [2];
- *(Leader selection)* Which UAVs in a multi-UAV system do we need to choose as leaders so the system can complete a surveillance task despite communication noise? [3].

The effort to answer such questions has culminated in a plethora of papers on topics such as actuator placement for controllability [4]–[9]; sensor scheduling for target tracking [1], [10]–[14]; and visual cue selection for robotic navigation [15], [16]. Notably, in all aforementioned papers the underlying optimization problem is of the form

$$\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} \quad f(\mathcal{A}), \tag{1}$$

where the set function f exhibits monotonicity and submodularity, a diminishing returns property [1]–[17]. In words, Problem (1) aims to find a set $\mathcal{A}$ of α elements from the finite ground set $\mathcal{V}$, such that $\mathcal{A}$ maximizes f. This problem is NP-hard, yet several good approximation algorithms have been proposed for its solution, such as the greedy [18].

[1]The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6228 USA (email: {vtzoumas, kgatsis, pappasg}@seas.upenn.edu).

[2]The author is with the Institute for Data, Systems and Society, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (email: jadbabai@mit.edu).

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and in part by AFOSR Complex Networks Program.

But sensors and actuators fail [19]; features can become obsolete [20]; and leaders can be attacked [21]. For example, sensors may fail due to malfunctions or adversarial attacks. In such scenarios, questions such as the following arise:

- Where to place a few actuators in a system, when some of them may fail? [22]
- Which sensors to activate to track an adversarial target that can jam a fraction of the activated sensors? [23], [24]
- Or, which features to select to train a robust machine learning model to changing features? [25]

In such scenarios, the optimization problem we need to address takes the form

$$\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} \min_{\mathcal{B}\subseteq\mathcal{A},|\mathcal{B}|\leq\beta} \quad f(\mathcal{A}\setminus\mathcal{B}), \tag{2}$$

where $\beta\leq\alpha$, which is a resilient formulation of Problem (1). In words, Problem (2) aims to find a set $\mathcal{A}$ of α elements such that $\mathcal{A}$ is resilient against the worst possible removal of β of its elements. Importantly, this formulation is suitable when we have no prior on the failure or attack mechanism.

The most relevant papers on the resilient monotone submodular optimization Problem (2) are [19] and [26]. In particular, Problem (2) was introduced in [19], where the authors proposed an approximation algorithm for the more general problem $\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha}\min_{i\in\{1,2,\ldots,m\}} f_i(\mathcal{A})$, where each f_i is monotone submodular. In more detail, the algorithm in [19] guarantees a high value for Problem (2) by allowing sets $\mathcal{A}$ up to size $\alpha[1+2\log(\alpha\beta\log(|\mathcal{V}|))]$, instead of α. Nonetheless, it runs with $O(|\mathcal{V}|^2(\frac{\alpha}{\beta})^\beta)$ evaluations of f, which is exponential in the number of possible removals β, and quadratic in the number of available elements for resiliency $|\mathcal{V}|$. This limits its applicability in large-scale settings where both β and $|\mathcal{V}|$ can be in the order of several thousands [27]. In [26], and its arxiv version [28], the authors prove that Problem (2) is NP-hard. In addition, they provide an algorithm for Problem (2) that runs with $O(|\mathcal{V}|(\alpha-\beta))$ evaluations of f, and which, for $\alpha,\beta\longrightarrow+\infty$, guarantees an approximate value at least $\simeq 29\%$ the optimal, the first approximation performance bound for Problem (2). However, in [26] the proposed algorithm is valid only when the number β of possible failures and attacks is up to $\sqrt{2\alpha}$, whereas in practice the number β of failures and attacks can be of the same order as the number α of selected sensors, actuators, etc. For example, it may be the case that β is up to $\alpha/2$ [29].

In this paper, we show how the notion of the curvature —deviation from modularity (additivity)— of a function can be used to provide a scalable algorithm for Problem (2) that has maximum resiliency, and for submodular functions with low curvature, superior approximation performance. Notably,

low curvature submodular functions are involved in a series of applications [11], [30], such as sensor placement for mutual information maximization [1]; feature selection for Gaussian process regression [11]; and active learning for speech recognition [16], [31].

In more detail, exploiting the curvature of the monotone submodular function f in Problem (2), denoted by κ_f, we provide an algorithm (Algorithm 1) with the properties:

- Algorithm 1 is valid for any number of selections for resiliency α, and any number of failures and attacks β;
- Algorithm 1 runs with $O(|\mathcal{V}|(\alpha-\beta))$ evaluations of f;
- Algorithm 1 guarantees the approximation bound $\frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right)\frac{1-\kappa_f}{2-\kappa_f}$.

Algorithm 1 improves upon the state-of-the-art algorithms for Problem (2) as follows:

- *High resiliency:* For $\beta > \sqrt{2\alpha}$, and any curvature values, Algorithm 1 is the first scalable algorithm for Problem (2) with bounded approximation performance.
- *High approximation performance:* For low curvature values, and any α and β, Algorithm 1 is the first scalable algorithm to exhibit approximation performance for Problem (2) up to at least 50% the optimal.

  For example, for the central problem in machine learning of Gaussian process regression with RBF kernels [1], [32], Algorithm 1 guarantees approximation performance at least 50% the optimal.

II. Resilient Submodular Maximization

We state the resilient maximization problem considered in this paper. To this end, we start with the definitions of monotone and submodular set functions.

Notation: For any set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ on a ground set $\mathcal{V}$, and any element $x \in \mathcal{V}$, $f(x)$ denotes $f(\{x\})$.

Definition 1 (Monotonicity): Consider any finite ground set $\mathcal{V}$. The set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is non-decreasing if and only if for any $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$, $f(\mathcal{A}) \leq f(\mathcal{A}')$. ◀

In words, a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is non-decreasing if and only if adding more elements in any $\mathcal{A} \subseteq \mathcal{V}$ cannot decrease the value of $f(\mathcal{A})$.

Definition 2 (Submodularity ([33], Proposition 2.1)): Consider any finite ground set $\mathcal{V}$. The set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is submodular if and only if

- for any $\mathcal{A} \subseteq \mathcal{V}$ and $\mathcal{A}' \subseteq \mathcal{V}$, $f(\mathcal{A}) + f(\mathcal{A}') \geq f(\mathcal{A} \cup \mathcal{A}') + f(\mathcal{A} \cap \mathcal{A}')$;
- equivalently, for any $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$ and $x \in \mathcal{V}$, $f(\mathcal{A} \cup \{x\}) - f(\mathcal{A}) \geq f(\mathcal{A}' \cup \{x\}) - f(\mathcal{A}')$. ◀

In words, a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is submodular if and only if it satisfies the following intuitive diminishing returns property: for any $x \in \mathcal{V}$, the gain $f(\mathcal{A} \cup \{x\}) - f(\mathcal{A})$ diminishes as $\mathcal{A}$ grows; equivalently, for any $\mathcal{A} \subseteq \mathcal{V}$ and $x \in \mathcal{V}$, the gain $f(\mathcal{A} \cup \{x\}) - f(\mathcal{A})$ is non-increasing.

In this paper, we consider the problem of resilient monotone submodular maximization, defined as follows.

Problem 1: Consider

- a finite ground set $\mathcal{V}$;
- a submodular and monotone set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that (without loss of generality) f is non-negative and $f(\emptyset) = 0$;
- and two integers α and β such that $0 \leq \beta \leq \alpha \leq |\mathcal{V}|$.

The problem of resilient monotone submodular maximization is to select a set $\mathcal{A}$ of α elements from the ground set $\mathcal{V}$, such that $f(\mathcal{A})$ is resilient against the worst possible removal $\mathcal{B}$ of β of $\mathcal{A}$'s elements. In mathematical notation,

$$\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} \min_{\mathcal{B}\subseteq\mathcal{A},|\mathcal{B}|\leq\beta} f(\mathcal{A}\setminus\mathcal{B}).$$ ◀

Problem 1 may be interpreted as a two-stage perfect information sequential game [34, Chapter 4], where the player that plays first chooses $\mathcal{A}$, and the player that plays second, knowing $\mathcal{A}$, chooses an optimal removal $\mathcal{B}$ from $\mathcal{A}$.

III. Algorithm for resilient submodular maximization

We present the first scalable algorithm for Problem 1, and show that it is valid for any number of failures and attacks, and that for functions with low curvature, guarantees superior approximation performance. To this end, we begin with the definition of the curvature of submodular functions.

A. Curvature of monotone submodular functions

We define the curvature of monotone submodular functions, which we use in our contributions in this paper. To this end, we start with the definition of modular set functions.

Definition 3 (Modularity): Consider any finite ground set $\mathcal{V}$. The set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is modular if and only if for any $\mathcal{A} \subseteq \mathcal{V}$, $f(\mathcal{A}) = \sum_{v\in\mathcal{A}} f(v)$. ◀

In words, a set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is modular if through f all elements in $\mathcal{V}$ cannot substitute each other, since Definition 3 implies that for any $\mathcal{A} \subseteq \mathcal{V}$ and $x \in \mathcal{V} \setminus \mathcal{A}$, $f(\{x\} \cup \mathcal{A}) - f(\mathcal{A}) = f(x)$; that is, in the presence of $\mathcal{A}$, x retains its contribution to the overall value of $f(\{x\} \cup \mathcal{A})$. In contrast, for a submodular set function $g : 2^{\mathcal{V}} \mapsto \mathbb{R}$, the elements in $\mathcal{V}$ *can* substitute each other, since Definition 2 implies $g(\{x\} \cup \mathcal{A}) - g(\mathcal{A}) \leq g(x)$; that is, in the presence of $\mathcal{A}$, x may lose part of its contribution to the overall value of $g(\{x\} \cup \mathcal{A})$.

Definition 4 (Curvature): Consider a finite ground set $\mathcal{V}$, and a monotone submodular set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that (without loss of generality) for $v \in \mathcal{V}$, $f(v) \neq 0$. The curvature of f is defined as

$$\kappa_f = 1 - \min_{v\in\mathcal{V}} \frac{f(\mathcal{V}) - f(\mathcal{V}\setminus\{v\})}{f(v)}.$$ ◀

In words, the curvature of a monotone submodular function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ measures how far f is from modularity: In particular, per Definition 2 of submodularity, it follows that curvature takes values $0 \leq \kappa_f \leq 1$, and

- $\kappa_f = 0$ if and only if for all $v \in \mathcal{V}$, $f(\mathcal{V}) - f(\mathcal{V} \setminus \{v\}) = f(v)$, that is, f is modular.
- $\kappa_f = 1$ if and only if there exist $v \in \mathcal{V}$ such that $f(\mathcal{V}) = f(\mathcal{V} \setminus \{v\})$, that is, in the presence of $\mathcal{V} \setminus \{v\}$, v loses all its contribution to the overall value of $f(\mathcal{V})$.

Algorithm 1 Algorithm for Problem 1.

Input: Per Problem 1, three are the inputs to Algorithm 1:

- finite set $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$, where $m = |\mathcal{V}|$;
- submodular and monotone set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is non-negative and $f(\emptyset) = 0$;
- and two integers α and β such that $0 \leq \beta \leq \alpha \leq |\mathcal{V}|$.

Output: Set $\mathcal{A}_{\text{RES}}$, with properties per Theorem 1.

1: $\mathcal{A}_1 \leftarrow \emptyset, \mathcal{A}_2 \leftarrow \emptyset$
2: Sort elements in $\mathcal{V}$ such that $\mathcal{V} = \{v'_1, v'_2, \ldots, v'_m\}$ and $f(v'_1) \geq f(v'_2) \geq \ldots \geq f(v'_m)$
3: $\mathcal{A}_1 \leftarrow \{v'_1, v'_2, \ldots, v'_\beta\}$
4: **while** $|\mathcal{A}_2| < \alpha - \beta$ **do**
5: $\quad x \in \arg\max_{y \in \mathcal{V} \setminus (\mathcal{A}_1 \cup \mathcal{A}_2)} [f(\{y\} \cup \mathcal{A}_2) - f(\mathcal{A}_2)]$
6: $\quad \mathcal{A}_2 \leftarrow \{x\} \cup \mathcal{A}_2$
7: **end while**
8: $\mathcal{A}_{\text{RES}} \leftarrow \mathcal{A}_1 \cup \mathcal{A}_2$

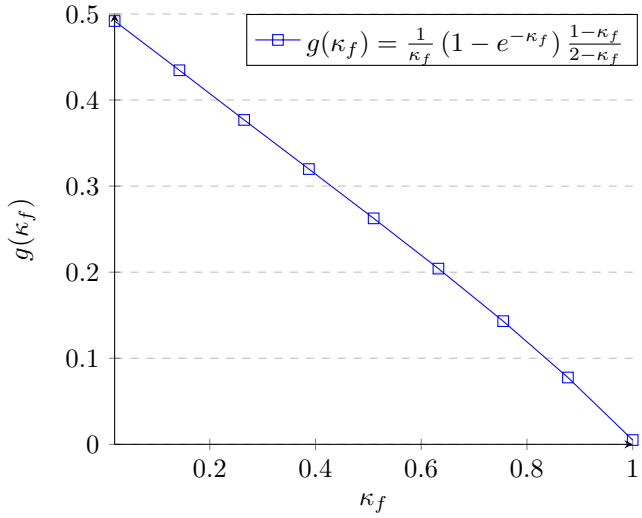

Fig. 1. Plot of $g(\kappa_f)$ versus curvature κ_f of a monotone submodular function f. By definition, the curvature κ_f of a monotone submodular function f takes values between 0 and 1. $g(\kappa_f)$ increases from 0 to 0.5 as κ_f decreases from 1 to 0.

An example of a monotone submodular function with zero curvature is the trace of the controllability matrix, which captures the control effort to drive the system in the state space [5]. A function with curvature 1 is the matroid rank function [35]. At the same time, many practically interesting functions have curvature strictly smaller than 1, such as the concave over modular functions [35, Section 2.1], and the $\log\det$ of positive-definite matrices [11], [36], which are used in applications such as speech processing [37], computer vision [38], feature selection for Gaussian process regression [1], and sensor scheduling for target tracking [13].

The notion of the curvature has served to tighten bounds for several submodular maximization problems, e.g., for the non-resilient optimization problem $\max_{\mathcal{A} \subseteq \mathcal{V}, |\mathcal{A}| \leq \alpha} f(\mathcal{A})$ the approximation bound of the greedy is tightened from $1 - 1/e$ to $\frac{1}{\kappa_f}(1 - e^{-\kappa_f})$ [35], [39], [40]. Nonetheless, for resilient submodular maximization problems, such as Problem 1, the role of the curvature has not been addressed yet. We provide the first results to this end in the next section.

B. Algorithm for resilient submodular maximization

We exploit the notion of the curvature to provide for the resilient maximization Problem 1 the Algorithm 1. In particular, Algorithm 1 returns a solution for Problem 1, denoted by $\mathcal{A}_{\text{RES}}$, in two steps: First, in lines 1-3 Algorithm 1 selects a set $\mathcal{A}_1$, which is composed of β elements from $\mathcal{V}$. Specifically, per line 2, each element in $v \in \mathcal{A}_1$ is such that for all $v' \in \mathcal{V} \setminus \mathcal{A}_1$, $f(v) \geq f(v')$. Second, in lines 4-8, Algorithm 1 selects greedily from $\mathcal{V} \setminus \mathcal{A}_1$ a set $\mathcal{A}_2$, which is composed of $\alpha - \beta$ elements, and then, in line 8, Algorithm 1 returns set $\mathcal{A}_{\text{RES}}$ as the union of $\mathcal{A}_1$ and $\mathcal{A}_2$.

Algorithm 1's performance is quantified in Theorem 1, whose proof can be found in the full version of this paper, located at the authors' websites. The intuition behind Algorithm 1 is discussed in Section III-C.

Theorem 1: Per Problem 1, consider

- a finite ground set $\mathcal{V}$;
- a submodular and monotone set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that f is non-negative and $f(\emptyset) = 0$;
- two integers α and β such that $0 \leq \beta \leq \alpha \leq |\mathcal{V}|$;
- $f^\star$ equal to the (optimal) value of Problem 1, i.e., $f^\star = \max_{\mathcal{A} \subseteq \mathcal{V}, |\mathcal{A}| \leq \alpha} \min_{\mathcal{B} \subseteq \mathcal{A}, |\mathcal{B}| \leq \beta} f(\mathcal{A} \setminus \mathcal{B})$;
- and for any $\mathcal{A} \subseteq \mathcal{V}$, $\mathcal{B}^\star(\mathcal{A})$ equal to the optimal removal of β elements from $\mathcal{A}$, i.e., $\mathcal{B}^\star(\mathcal{A}) \in \min_{\mathcal{B} \subseteq \mathcal{A}, |\mathcal{B}| \leq \beta} f(\mathcal{A} \setminus \mathcal{B})$.

The following two hold on Algorithm 1's performance:

1) Algorithm 1 returns $\mathcal{A}_{\text{RES}} \subseteq \mathcal{V}$ such that $|\mathcal{A}_{\text{RES}}| \leq \alpha$, and for $\kappa_f \neq 0$,
$$f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}})) \geq \frac{1}{\kappa_f}\left(1 - e^{-\kappa_f}\right)\frac{1 - \kappa_f}{2 - \kappa_f} f^\star;$$
in addition, for $\kappa_f = 0$, $f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}})) = f^\star$.
2) Algorithm 1 runs in $O(|\mathcal{V}|(\alpha - \beta))$ evaluations of f. ◀

Remark 1: Given any finite ground set $\mathcal{V}$ and monotone submodular function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ in the resilient maximization Problem 1, the approximation bound of Algorithm 1 depends on the curvature value which is computed with $O(|\mathcal{V}|)$ evaluations of f, per Definition 4. ◀

Remark 2: Per Fig. 1 and Theorem 1, the worst-case approximation performance bound of Algorithm 1 can be approximated as $\frac{1}{\kappa_f}(1 - e^{-\kappa_f})\frac{1-\kappa_f}{2-\kappa_f} \simeq -0.5\kappa_f + 0.5$. ◀

Remark 3: For zero curvature, Algorithm 1 solves Problem 1 exactly. At the same time, for non-zero curvature, in which case Problem 1 is NP-hard [28], Algorithm 1's approximation bound is $\frac{1}{\kappa_f}(1 - e^{-\kappa_f})\frac{1-\kappa_f}{2-\kappa_f}$, which tends to 0.5 as $\kappa_f \longrightarrow 0$. This discontinuity between the exact solution performance of Algorithm 1 for zero curvature, and its at least 50% the optimal performance for $\kappa_f \longrightarrow 0$ suggests that in Theorem 1 Algorithm 1's approximation bound $\frac{1}{\kappa_f}(1 - e^{-\kappa_f})\frac{1-\kappa_f}{2-\kappa_f}$ for $\kappa_f \neq 0$ can be tightened. The same observation is supported by the proof for Theorem 1, where we consider two cases with respect to the properties of the optimal removal $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$ and only for the one of them we are able to show the tighter bound

$\frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right)$ —which indeed tends to 1 as $\kappa_f \longrightarrow 0$— while for the other case we employ a looser bounding methodology that leads to the bound $\frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right)\frac{1-\kappa_f}{2-\kappa_f}$ in Theorem 1. ◀

Theorem 1 implies for Algorithm 1's performance:

- Algorithm 1 is the first scalable algorithm for Problem 1 that is valid for any number of failures and attacks β, and any number of selections for resiliency α. In particular, the previously proposed algorithms for Problem 1 in [19] and [26] are such that: The algorithm in [19] runs in exponential time in β, and quadratic in $\mathcal{V}$, and as a result, has limited applicability in large-scale settings. The algorithm in [26] is valid only for $\beta \leq \sqrt{2\alpha}$, and as a result, has limited applicability in decision and control settings where the number of failures and attacks β can be up to the number of placed sensors, actuators, etc. α. For example, the inequality $\beta \leq \sqrt{2\alpha}$ is violated when among 100 placed sensors, 20 may fail or be attacked.
- Algorithm 1 is the first scalable algorithm for Problem 1 that for non-zero curvature values $\kappa_f \leq 0.42$, and any number of failures and attacks β, guarantees approximation performance more than at least 30% the optimal. In particular, the previously proposed algorithms for Problem 1 in [19] and [26] are such that: As mentioned above, the algorithm in [19] runs in exponential time in β, and quadratic in $\mathcal{V}$, and as a result, has limited applicability in large-scale settings. And the algorithm in [26], under the constraint $\beta \leq \sqrt{2\alpha}$, and for $\alpha, \beta \longrightarrow +\infty$, guarantees an approximate value up to at least 29% the optimal. In particular, for $\alpha, \beta < +\infty$, its approximation performance is less than at least 29% the optimal.

An example of a central problem in machine learning, with applications, e.g., in sensor placement, where Algorithm 1 guarantees approximation performance at least $\simeq 50\%$ the optimal is that of Gaussian process regression for Gaussian processes with RBF kernels [1], [32]. The reason is that in this class of problems the objective function is the entropy of the selected measurements, which for Gaussian processes with RBF kernels was shown recently to have curvature values close to zero [11, Theorem 5].

Overall, Algorithm 1's curvature-dependent approximation bound makes a first step towards separating the class of monotone submodular functions into the functions for which the resilient maximization Problem 1 can be approximated well (low curvature functions), and the functions for which it cannot (high curvature functions). The reason is that Algorithm 1's approximation bound increases as the curvature decreases, and for zero curvature it becomes 1 —i.e., for zero curvature, Algorithm 1 solves Problem 1 exactly. This role of the curvature in Problem 1 is similar to the role that the curvature has played in the non-resilient variant of Problem 1, i.e., the optimization problem $\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} f(\mathcal{A})$, where it has been used to separate the class of submodular functions into the functions for which $\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} f(\mathcal{A})$ can be approximated well (low curvature functions), and the functions for which it cannot (high curvature functions) [35], [39]. Hence, Theorem 1 also supports the intuition that the resilient maximization Problem 1 is easier when the non-resilient variant $\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} f(\mathcal{A})$ is easy, and it is harder when $\max_{\mathcal{A}\subseteq\mathcal{V},|\mathcal{A}|\leq\alpha} f(\mathcal{A})$ is hard.

C. Intuition behind algorithm for resilient maximization

We explain the intuition behind Algorithm 1 for Problem 1, and give also an illustrative example. To this end, we focus only on the NP-hard case where the curvature of Problem 1's objective function is non-zero [28, Lemma 3].

We explain how Algorithm 1 works first for the case $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_1$, and then for the case $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) \neq \mathcal{A}_1$. Notably, the case $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_1$ is possible since in lines 1-3 Algorithm 1 selects a set $\mathcal{A}_1$ such that $|\mathcal{A}_1| = \beta$, and per Problem 1, β is the number of possible removals.

a) Intuition behind Algorithm 1 for $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_1$: The two parts of Algorithm 1 operate in tandem to guarantee

$$f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}})) \geq \frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right) f^\star. \tag{3}$$

This happens as follows: $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_1$ implies that $\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_2$, since $\mathcal{A}_{\text{RES}} = \mathcal{A}_1 \cup \mathcal{A}_2$, per line 8 of Algorithm 1. Therefore, $f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}})) = f(\mathcal{A}_2)$. But in lines 4-7 of Algorithm 1, $\mathcal{A}_2$ is selected greedily, and as a result, [39, Theorem 5.4]

$$f(\mathcal{A}_2) \geq \frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right) \max_{\mathcal{A}\subseteq\mathcal{V}\setminus\mathcal{A}_1,|\mathcal{A}|\leq\alpha-\beta} f(\mathcal{A}).$$

Finally, $\max_{\mathcal{A}\subseteq\mathcal{V}\setminus\mathcal{A}_1,|\mathcal{A}|\leq\alpha-\beta} f(\mathcal{A}) \geq f^\star$, since the left-hand-side of this inequality is the maximum value one can achieve for f by choosing $\alpha-\beta$ elements from $\mathcal{V}$ *knowing* that $\mathcal{A}_1$ has been removed from $\mathcal{V}$, whereas the right-hand-side is the maximum value one can achieve for f by choosing α elements from $\mathcal{V}$ *not knowing* which β of them will be optimally removed; a mathematical version of the latter proof can be found in [28, Lemma 2].

b) Intuition behind Algorithm 1 for $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) \neq \mathcal{A}_1$: The two parts of Algorithm 1 operate in tandem to guarantee,

$$f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}})) \geq \frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right)\frac{1-\kappa_f}{2-\kappa_f} f^\star. \tag{4}$$

This happens as follows: $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) \neq \mathcal{A}_1$ implies, along with $|\mathcal{B}^\star(\mathcal{A}_{\text{RES}})| = |\mathcal{A}_1| = \beta$, that if μ elements in $\mathcal{A}_1$ are *not* included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$, exactly μ elements in $\mathcal{A}_2$ are included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$. Therefore, $f(\mathcal{A}_{\text{RES}} \setminus \mathcal{B}^\star(\mathcal{A}_{\text{RES}}))$ can take a bounded value similar to (3) only if the μ elements in $\mathcal{A}_1$ that are *not* included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$ can *compensate* for the μ elements in $\mathcal{A}_2$ that are included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$. For this reason, in line 2, Algorithm 1 chooses the elements in $\mathcal{A}_1$ so that they a have higher value than those in $\mathcal{V}\setminus\mathcal{A}_1$: In particular, using the fact that the elements in $\mathcal{A}_1$ have a higher value than those in $\mathcal{V}\setminus\mathcal{A}_1$, and the definition of the curvature κ_f, in the proof of Theorem 1 we bound how much value the elements in $\mathcal{A}_1$ that are *not* included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$ can compensate for the value of the elements in $\mathcal{A}_2$ that are included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$, and conclude (4).

Overall, in the worst-case Algorithm 1 guarantees for the resilient maximization Problem 1 the approximation

performance bound (4), as stated in Theorem 1, since for all values of the curvature κ_f, the bound in (4) is smaller than the bound in (3), i.e., $\frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right) \geq \frac{1}{\kappa_f}\left(1-e^{-\kappa_f}\right)\frac{1-\kappa_f}{2-\kappa_f}$, and we do not know a priori which of the two preceding bounds holds at a problem instance.

Example 1: We use an instance of Problem 1 to illustrate how Algorithm 1 finds an approximate solution to Problem 1, as well as, how it performs. We consider the following instance: Let $\alpha = 2$, $\beta = 1$, $\mathcal{V} = \{v_1, v_2, v_3\}$, and f such that $f(\emptyset) = 0$, $f(v_3) > 0$, $f(v_1) = f(v_3) + 1$, $f(v_2) = f(v_3) + 0.5$, $f(\{v_1\} \cup \{v_2\}) = f(v_3) + 1$, $f(\{v_1\} \cup \{v_3\}) = f(\mathcal{V}) = 2f(v_3) + 1$, and $f(\{v_2\} \cup \{v_3\}) = 2f(v_3) + 0.5$.

For the aforementioned instance, the curvature is $\kappa_f = 1$, and as a result, Algorithm 1 is guaranteed to return $\mathcal{A}_{\text{RES}}$ such that the approximation ratio is either at least $1 - 1/e$, per bound (3), or at least 0, per bound (4).

Algorithm 1 selects $\mathcal{A}_{\text{RES}} = \{v_1, v_2\}$, which in this example is the exact solution to Problem 1. In particular, in lines 2-3, Algorithm 1 selects $\mathcal{A}_1 = \{v_1\}$, and in lines 4-7, it selects $\mathcal{A}_2 = \{v_2\}$. The optimal removal is $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \{v_1\}$.

The reason that Algorithm 1 performs optimally in this example, which is not expected by Theorem 1 since it has the worst curvature value $\kappa_f = 1$, is as follows: In lines 1-3 Algorithm 1 selects $\mathcal{A}_1 = \{v_1\}$, which for $\mathcal{A}_{\text{RES}} = \{v_1, v_2\}$ is the element that will be included in the optimal removal $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$. That is, in this example $\mathcal{B}^\star(\mathcal{A}_{\text{RES}}) = \mathcal{A}_1$, which implies that the approximation performance of Algorithm 1 is bounded by $1 - 1/e$, as in (3). This is the first important observation towards explaining the optimal performance of Algorithm 1 in this example; the second and final necessary observation is as follows: In lines 4-7, Algorithm 1 selects the best element in $\mathcal{V}$ assuming that the element in $\mathcal{A}_1$ will be included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$, i.e., removed from $\mathcal{A}_{\text{RES}}$, since it selects greedily from $\mathcal{V} \setminus \mathcal{A}_1$. In contrast, if in lines 4-7 Algorithm 1 would select greedily from $\mathcal{V}$ without taking into account that the element in $\mathcal{A}_1$ is going to be included in $\mathcal{B}^\star(\mathcal{A}_{\text{RES}})$, then it would select $\mathcal{A}_{\text{RES}} = \{v_1, v_3\}$ which is suboptimal for Problem 1. ◀

IV. Simulations

We empirically test Algorithm 1's approximation performance for Problem 1 against an exact, brute force algorithm. As a test function, with consider one of the following form: Given a finite ground set $\mathcal{V}$, and $|\mathcal{V}|$ positive semi-definite matrices $D_1, D_2, \ldots, D_{|\mathcal{V}|}$, for any $\mathcal{A} \subseteq \mathcal{V}$, let $f(\mathcal{A}) = \log\det(\sum_{i\in\mathcal{A}} D_i + I)$, where I is the identity matrix. Functions of this form appear in applications such as sensor selection for Gaussian process regression [1], and actuator placement for bounded control effort [7], [41]. To run our simulations, and be able to compute the exact value to Problem 1, we select small sizes of $|\mathcal{V}|$ from 8 to 15. In addition, we fix the number of selections for resiliency α to 7, vary the number of attacks/failures β from 1 to 6, and for each of the aforementioned cases, generate 10 random instances of the matrices $D_1, D_2, \ldots, D_{|\mathcal{V}|}$ of size 20×20.

Our simulations are summarized in Fig. 2, where Algorithm 1 is seen to perform close to 98% the optimal, and its approximation performance to degrade as β increases up to α, which is equal to 7. Notably, for all generated instances of f, f's curvature takes values larger than 0.9, for which, Algorithm 1's worst-case theoretical approximation bound in Theorem 1 is less than 10%. Overall, Algorithm 1's approximation performance in Fig. 2 is in accordance with the empirical observation that greedy-type algorithms for submodular maximization often outperform in practice their worst-case theoretical approximation guarantees [1], [11].

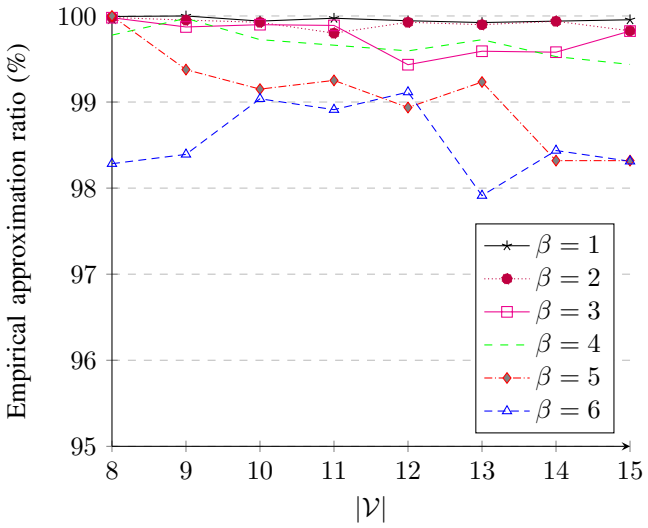

Fig. 2. Empirical approximation performance of Algorithm 1. For details, see Section IV.

V. Concluding Remarks & Future Work

We focused on the resilient submodular maximization Problem 1, which is central in machine learning, optimization, and control, in applications such as feature selection for classifier training, and sensor scheduling for target tracking. In particular, exploiting the notion of curvature, we provided the first scalable algorithm for Problem 1, Algorithm 1, which is valid for any number of attacks or failures, and which, for functions with low curvature, guarantees superior approximation performance. In addition, for functions with zero curvature, Algorithm 1 solves Problem 1 exactly. Overall, Algorithm 1's approximation bound makes a first step to characterize the curvature's effect on approximations for resilient submodular maximization problems, complementing that way the current knowledge on the curvature's effect on *non-resilient* submodular maximization. Future work will focus on Algorithm 1's extension to matroid constraints.

References

[1] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J.l of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[2] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 1057–1064.

[3] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, *Submodularity in dynamics and control of networked systems*. Springer, 2016.

[4] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[5] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, March 2014.

[6] S. Pequito, S. Kar, and A. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *IEEE Trans. on Automatic Control*, vol. 61, no. 2, pp. 303–318, 2016.

[7] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2016.

[8] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. on Control of Network Systems*, vol. 3, no. 1, pp. 67–78, 2016.

[9] E. Nozari, F. Pasqualetti, and J. Cortes, "Time-varying actuator scheduling in complex networks," *arXiv preprint:1611.06485*, 2016.

[10] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.

[11] D. Sharma, A. Kapoor, and A. Deshpande, "On greedy maximization of entropy," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1330–1338.

[12] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Near-optimal sensor scheduling for batch state estimation," in *IEEE 55th Conference on Decision and Control*, 2016, pp. 2695–2702.

[13] V. Tzoumas, N. A. Atanasov, A. Jadbabaie, and G. J. Pappas, "Scheduling nonlinear sensors for stochastic process estimation," in *IEEE American Control Conference*, 2017, to appear.

[14] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202 – 210, 2017.

[15] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *arXiv preprint:1610.03344*, 2016.

[16] R. Iyer, S. Jegelka, and J. Bilmes, "Fast semidifferential-based submodular function optimization," in *Proceedings of the 30th International Conference on International Conference on Machine Learning*, 2013, pp. 855–863.

[17] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, November 2008.

[18] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Mathematics of operations research*, vol. 3, no. 3, pp. 177–188, 1978.

[19] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.

[20] A. Globerson and S. Roweis, "Nightmare at test time: Robust learning by feature deletion," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 353–360.

[21] A. Clark, L. Bushnell, and R. Poovendran, "Leader selection games under link noise injection attacks," in *Proceedings of the 1st International Conf. on High Confidence Networked Systems*, 2012, pp. 31–40.

[22] S. Pequito, G. Ramos, S. Kar, A. P. Aguiar, and J. Ramos, "On the Exact Solution of the Minimal Controllability Problem," *arXiv preprint:1401.4209*, 2014.

[23] A. Das and D. Kempe, "Sensor selection for minimizing worst-case prediction error," in *Proceedings of the 7th International conference on Information processing in sensor networks*, 2008, pp. 97–108.

[24] A. Laszka, Y. Vorobeychik, and X. Koutsoukos, "Resilient observation selection in adversarial settings," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 7416–7421.

[25] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks," *IEEE Network*, vol. 20, no. 3, pp. 41–47, 2006.

[26] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," in *18th International Conf. on Integer Programming and Combinatorial Optimization*, 2016, pp. 312–324.

[27] Y. Shoukry, M. Chong, M. Wakaiki, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, J. P. Hespanha, and P. Tabuada, "SMT-based observer design for cyber-physical systems under sensor attacks," in *Proc. of the 7th International Conf. on Cyber-Physical Systems*, 2016.

[28] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," *arXiv preprint:1507.06616*, 2015.

[29] M. S. Chong, M. Wakaiki, and J. P. Hespanha, "Observability of linear systems under adversarial attacks," in *American Control Conference*, 2015, pp. 2439–2444.

[30] S. Jegelka, "Combinatorial problems with submodular coupling in machine learning and computer vision," Ph.D. dissertation, ETH Zurich, 2012.

[31] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *10th Annual Conf. of the International Speech Comm. Association.*, 2009.

[32] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

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

[34] R. B. Myerson, *Game theory: Analysis of conflict*. Harvard University Press, 2013.

[35] R. K. Iyer, S. Jegelka, and J. A. Bilmes, "Curvature and optimal algorithms for learning and minimizing submodular functions," in *Advances in Neural Inform. Processing Systems*, 2013, pp. 2742–2750.

[36] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," in *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015, pp. 1134–1148.

[37] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*, 2011, pp. 510–520.

[38] S. Jegelka and J. Bilmes, "Submodularity beyond submodular energies: Coupling edges in graph cuts," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1897–1904.

[39] 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.

[40] J. Vondrák, "Submodularity and curvature: The optimal algorithm," *RIMS Kokyuroku Bessatsu*, vol. 23, pp. 253–266, 2010.

[41] V. Tzoumas, M. Rahimian, G. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *arXiv preprint:1409.3289*, 2014.