Robust and Adaptive Sequential Submodular Optimization

Vasileios Tzoumas, *Member, IEEE*, Ali Jadbabaie, *Fellow, IEEE*, and George J. Pappas, *Fellow, IEEE*

***Abstract*—Emerging applications of control, estimation, and machine learning, from target tracking to decentralized model fitting, pose resource constraints that limit which of the available sensors, actuators, or data can be simultaneously used across time. Therefore, many researchers have proposed solutions within discrete optimization frameworks where the optimization is performed over finite sets. By exploiting notions of discrete convexity, such as submodularity, the researchers have been able to provide scalable algorithms with provable suboptimality bounds. In this article, we consider such problems but in adversarial environments, where in every step a number of the chosen elements in the optimization is removed due to failures/attacks. Specifically, we consider for the first time a sequential version of the problem that allows us to observe the failures and adapt, while the attacker also adapts to our response. We call the novel problem *robust sequential submodular maximization* (RSM). Generally, the problem is computationally hard and no scalable algorithm is known for its solution. However, in this article, we propose robust and adaptive maximization (RAM), the first scalable algorithm. RAM runs in an online fashion, adapting in every step to the history of failures. Also, it guarantees a near-optimal performance, even against any number of failures among the used elements. Particularly, RAM has both provable per-instance *a priori* bounds and tight and/or optimal *a posteriori* bounds. Finally, we demonstrate RAM's near-optimality in simulations across various application scenarios, along with its robustness against several failure types, from worst-case to random.**

***Index Terms*—Algorithm design and analysis, approximation algorithms, autonomous systems, combinatorial mathematics, control, estimation, fault-tolerant systems, mobile robots, multirobot systems.**

I. Introduction

CONTROL, estimation, and machine learning applications of the Internet of Things and autonomous robots [1] require the sequential optimization of systems in scenarios such as follows.

1) *Sensor scheduling:* An unmanned aerial vehicle (UAV) is assisted for its navigation by on-board and on-ground sensors. Ideally, the UAV would use all available sensors for navigation. However, limited on-board capacity for measurement-processing necessitates a sequential sensor scheduling problem [2]: at each time step, which few sensors should be used for the UAV to effectively navigate itself?
2) *Target tracking:* A wireless sensor network (WSN) is designated to monitor a mobile target. Limited battery power necessitates a sequential sensor activation problem [3]: at each time step, which few sensors should be activated for the WSN to effectively track the target?
3) *Decentralized model fitting:* A team of mobile robots collects data to learn the model of an unknown environmental process. The data are transmitted to a fusion center, performing the statistical analysis. Ideally, all robots would transmit their data to the center at the same time. But instead, communication bandwidth constraints necessitate a sequential transmission problem [4]: at each time step, which few robots should transmit their data for the center to effectively learn the model?

Similar applications of sensor and data scheduling, but also of actuator scheduling as well as infrastructure design are studied in [5]–[16]. Particularly, all applications mentioned above require the sequential selection of a few elements, among a finite set of available ones, to optimize performance across multiple steps subject to resource constraints. For example, the target tracking application mentioned above requires the sequential activation of a few sensors across the WSN, to optimize an estimation error subject to power constraints. Importantly, the activated sensors may vary in time, since each sensor may measure different parts of the target's state (e.g., some sensors may measure only position, others only speed). Formally, all

Manuscript received May 8, 2020; revised October 12, 2020; accepted December 13, 2020. Date of publication December 21, 2020; date of current version December 29, 2021. This work was supported in part by the AFOSR Complex Networks program and in part by the Army Research Laboratory Collaborative Research Alliance under Distributed and Collaborative Intelligent Systems and Technology under Grant W911NF-17-2-0181 program. Recommended by Associate Editor F. Pasqualetti. *(Corresponding author: Vasileios Tzoumas.)*

Vasileios Tzoumas is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109 USA, and also with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: vtzoumas@umich.edu).

Ali Jadbabaie is with the Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jadbabai@mit.edu).

George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: pappasg@seas.upenn.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2020.3046222.

Digital Object Identifier 10.1109/TAC.2020.3046222

0018-9286 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

abovementioned applications motivate the sequential optimization problem[1]

$$\max_{\mathcal{A}_1\subseteq\mathcal{V}_1}\cdots\max_{\mathcal{A}_T\subseteq\mathcal{V}_T} f(\mathcal{A}_1,\ldots,\mathcal{A}_T) \quad \text{s.t.} \quad |\mathcal{A}_t| = \alpha_t, \quad t=1,\ldots,T \tag{1}$$

where T is a given horizon; $\mathcal{V}_t$ is a given finite set of available elements to choose from at t such that $\mathcal{V}_t\cap\mathcal{V}_{t'}=\emptyset$ for all $t,t'=1,\ldots,T$;[2] $f:2^{\mathcal{V}_1}\cup\cdots\cup 2^{\mathcal{V}_T}\mapsto\mathbb{R}$ is a given objective function; α_t is a given cardinality constraint, capturing the resource constraints at t; and $\mathcal{A}_t$ are the chosen elements at t, resulting from the solution of (1). Notably, in all abovementioned applications, and [5]–[16], f is nondecreasing, and without loss of generality one may consider $f(\emptyset)=0$. For example, in [11], f is the trace of the inverse of the controllability Gramian, which captures the average control effort for driving the system; and in [8], f is the logdet of the error covariance of the minimum mean square batch-state estimator. Specifically, in [8], f is also submodular, a diminishing returns property that captures the intuition that a sensor's contribution to fs value diminishes when more sensors are activated already.

Although the problem in (1) is computationally hard, efficient algorithms have been proposed for its solution: when f is monotone and submodular, then (1) is NP-hard [17] and the greedy algorithm in [18, Sec. 4] guarantees a constant suboptimality bound across all problem instances; and when f is only monotone, then (1) is inapproximable (no polynomial time algorithm guarantees a constant bound across all instances) [19], [20] but the greedy algorithm in [18] guarantees per-instance bounds instead [21]–[23].

In this article, however, we shift focus to a novel reformulation of (1) that is robust against failures/attacks. Particularly, in all abovementioned applications, at any time t, actuators can be cyber-attacked [24], sensors can malfunction [25], and communication channels can be blocked [4], all resulting to denial-of-service (DoS) failures, in the sense that the actuators, sensors, channels, etc., will shut down (stop working), at least temporarily. Hence, in such failure-prone and adversarial scenarios, (1) may fail to protect any of the abovementioned applications, since it ignores the possibility of DoS failures. Thus, toward guaranteed protection, a robust reformulation becomes necessary *that can both adapt to the history of incured failures and account for future ones*.

Therefore, in this article, we introduce a novel robust optimization framework, named *robust sequential submodular maximization* (RSM) that goes beyond the failure-free (1) and accounts for DoS failures/attacks. Specifically, we define RSM as the following robust reformulation of (1):

RSM problem:

$$\max_{\mathcal{A}_1\subseteq\mathcal{V}_1}\min_{\mathcal{B}_1\subseteq\mathcal{A}_1}\cdots\max_{\mathcal{A}_T\subseteq\mathcal{V}_T}\min_{\mathcal{B}_T\subseteq\mathcal{A}_T} f(\mathcal{A}_1\setminus\mathcal{B}_1,\ldots,\mathcal{A}_T\setminus\mathcal{B}_T) \quad \text{s.t.} \quad |\mathcal{A}_t| = \alpha_t,\ |\mathcal{B}_t|\leq\beta_t, \quad t=1,\ldots,T \tag{2}$$

where β_t is a given number of possible failures (generally, $\beta_t\in[0,\alpha_t]$); and $\mathcal{B}_t$ is the failure against $\mathcal{A}_t$.

By solving RSM, our goal is to maximize f despite worst-case failures that occur at each maximization step, as captured by the intermediate/subsequent minimization steps. Evidently, since RSM considers worst-case failures, it is suitable when there is no prior on the failure mechanism, or when protection against worst-case failures is essential, such as in safety-critical target tracking and costly experiment designs.

RSM can be interpreted as a T-stage perfect information sequential game between a "maximization" player (defender) and a "minimization" player (attacker) [26, Ch. 4]. The defender starts the game and the players act sequentially, having perfect knowledge of each others' actions: at each t, the defender selects an $\mathcal{A}_t$, and then the attacker responds with a worst-case removal $\mathcal{B}_t$ from $\mathcal{A}_t$, while both players account for the history of all actions up to $t-1$. In this context, the defender finds an optimal sequence $\mathcal{A}_1,\ldots,\mathcal{A}_T$ by accounting at each t

1) for the history of responses $\mathcal{B}_1,\ldots,\mathcal{B}_{t-1}$;
2) for the subsequent response $\mathcal{B}_t$;
3) for all remaining future responses $\mathcal{B}_{t+1},\ldots,\mathcal{B}_T$.

This is an additional computational challenge in comparison to the failure-free (1), which is already computationally hard.

No scalable algorithms exist for RSM. *In this article, to provide the first scalable algorithm, we develop an adaptive algorithm* that at each t accounts only (i) for the history of responses up to $t-1$ and (ii) for the subsequent response $\mathcal{B}_t$ (but not for the remaining future responses up to $t=T$), and as a result is scalable, but which still can guarantee a performance close to the optimal.

Related work in combinatorial optimization: The majority of the related work has focused on the failure-free (1), when f is either monotone and submodular or only monotone. In more detail, Fisher *et al.* [18] focused on f being monotone and submodular, and proposed offline and online greedy algorithms that both guarantee the constant $1/2$ suboptimality bound. Similarly, Conforti and Cornuéjols [27], Iyer *et al.* [28], and Sviridenko *et al.* [23] focused again on f being monotone and sumodular but provided instead per-instance, curvature-depended bounds. The bounds generally tighten the ones in [18]. Finally, Krause *et al.* [29], Das and Kempe [21], Wang *et al.* [22], and Sviridenko *et al.* [23] (see also the earlier [30]) focused on f being only monotone, and proved per-instance, curvature-depended bounds for the greedy algorithms in [18], using notions of curvature—also referred to as "submodularity ratio"—they introduced.

Recent work has also studied failure-robust reformulations of (1), typically per RSM's framework *but only for $T=1$, where no adaptiveness is required*. Specifically, when f is monotone

[1]Calligraphic fonts denote finite discrete sets (e.g., $\mathcal{A}$). $2^{\mathcal{A}}$ denotes $\mathcal{A}$s power set. $|\mathcal{A}|$ its cardinality. $\mathcal{A}\setminus\mathcal{B}$ denotes set difference: the elements in $\mathcal{A}$ not in $\mathcal{B}$. Given a set function $f:2^{\mathcal{V}_1}\cup\cdots\cup 2^{\mathcal{V}_T}\mapsto\mathbb{R}$, and $\mathcal{A}_1\subseteq\mathcal{V}_1,\ldots,\mathcal{A}_t\subseteq\mathcal{V}_t$, for some positive integer $t\leq T$, the $f(\mathcal{A}_1,\ldots,\mathcal{A}_t)$ denotes $f(\mathcal{A}_1\cup\cdots\cup\mathcal{A}_t\cup\emptyset\cup\cdots\cup\emptyset)$ where the $\emptyset$ is repeated $T-t$ times, and $\emptyset$ denotes the empty set. $\mathbb{R}$ denotes the set of real numbers.

[2]Even if the elements in $\mathcal{V}_1,\ldots,\mathcal{V}_T$ correspond to the same system modules, e.g., sensors, the elements among different $\mathcal{V}_t$ are differentiated because they are chosen at different times. For example, consider the case where $T=2$, and two sensors s_1 and s_2 are available to be chosen at each t; then, by denoting with $s_{i,t}$ that sensor i is available to be chosen at t, it is $\mathcal{V}_1=\{s_{1,1},s_{2,1}\}$ and $\mathcal{V}_2=\{s_{1,2},s_{2,2}\}$, and, naturally, $\mathcal{V}_1\cap\mathcal{V}_2=\emptyset$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

and submodular, Orlin *et al.* [31] and Bogunovic *et al.* [32] provided greedy algorithms with constant suboptimality bounds. However, the algorithms are valid only for limited numbers of failures (for $\beta_1 \leq \sqrt{\alpha_1}$ in [31] and $\beta_1 \leq \alpha_1/(\log \alpha_1)^3$ in [32]). In contrast, Tzoumas *et al.* [33] provided a greedy algorithm with per-instance bounds for any number of failures (β_1 can take any value in $[0, \alpha_1]$). Also, Rahmattalabi *et al.* [34] developed a mixed-integer linear program approach for a locations monitoring problem. More recently, Tzoumas *et al.* [35] and Bogunovic *et al.* [36] extended the previous works on the $T = 1$ case by focusing on f being only monotone, and proved per-instance, curvature-dependent bounds for the algorithm introduced in [33]. In more detail, Bogunovic *et al.* [36] focus on cardinality constraints, whereas Tzoumas *et al.* [35] on the more general matroid constraints, but, still, for the case where $T = 1$. The latter framework enabled applications of failure-robust multirobot robot planning, and particularly of active information gathering [37] and target tracking [38]. Other relevant work is that of Mitrovic *et al.* [39], where a memoryless failure-robust reformulation of (1) is considered, instead of the sequential framework of RSM, which takes into account the history of past selections/failures. Finally, Mirzasoleiman *et al.* [40] and Kazemi *et al.* [41] adopted a robust optimization framework against non-worst-case failures, in contrast to RSM, which is against worst-case failures.

All in all, in comparison to all prior research, in this article, we analyze RSM's multistep case $T > 1$ for the first time, and consider adaptive algorithms.

Related work in control: In the robust/secure control literature, various approaches have been proposed toward fault-tolerant control, secure control, as well as secure state estimation, against random failures, data injection, and DoS failures/attacks [42]–[61]. In contrast to RSM's resource-constrained framework, the authors in[42]–[61] focus in resource abundant environments where all sensors and actuators stay always active under normal operation. For example, the authors in[59]–[61] focus on DoS failures/attacks from the perspective of packet loss and intermittent network connectivity, which can result to system destabilization. Generally, the authors in[42]–[61] focus on failure/attack detection and identification, and/or secure estimator/controller design, *instead of the adaptive activation of a few sensors/actuators against worst-case DoS failures/attacks per RSM*.

Contributions. We introduce the novel RSM problem of robust sequential maximization against DoS failures/attacks. We develop the first scalable algorithm, named robust and adaptive maximization (RAM), that has the following the properties.

1) *Adaptiveness:* At each time $t = 1, 2, \ldots$, RAM selects a robust solution $\mathcal{A}_t$ in an online fashion, accounting for the history of failures $\mathcal{B}_1, \ldots, \mathcal{B}_{t-1}$ and of actions $\mathcal{A}_1, \ldots, \mathcal{A}_{t-1}$, as well as, for all possible failures at t from $\mathcal{A}_t$.
2) *System-wide robustness:* RAM is valid for any number of failures; that is, for any $\beta_t \in [0, \alpha_t]$, $t = 1, 2, \ldots$.
3) *Polynomial running time:* RAM has the same order of running time as the polynomial time greedy algorithm proposed in [18, Sec. 4] for the failure-free (1).
4) *Provable approximation performance:* RAM has provable per-instance suboptimality bounds that quantify RAM's near-optimality at each problem instance at hand.[3]

Particularly, we provide both *a priori* and *a posteriori* per-instance bounds. The *a priori* bounds quantify RAM's near-optimality before RAM has run. In contrast, the *a posteriori* bounds are computable online (as RAM runs), once the failures at each current step have been observed. The *a posteriori* bounds are tight and/or optimal.[4] Finally, we present approximations of the *a posteriori* bounds that are computable before each failure occurs. To quantify the bounds, we use curvature notions by Conforti and Conruéjols [27], for monotone and submodular functions, and Sviridenko *et al.* [23], for monotone functions.

We demonstrate RAM's effectiveness in applications of sensor scheduling, and of target tracking with WSNs. We present a Monte Carlo analysis, where we vary the failure types from worst-case to greedily and randomly selected failures, and compare RAM against a brute-force optimal algorithm (viable only for small-scale instances), the greedy algorithm in [18], and a random algorithm. In the results, we observe RAM's near-optimality against worst-case failures, its robustness against non-worst-case failures, and its superior performance against the compared algorithms.

Comparison with the preliminary results in [62], *which coincides with preprint* [62]: This article extents the results in [62], considers new simulations, and includes the proofs that were all omitted from [62]. Particularly, most of the technical results herein, including Theorem 13, Theorem 14, Corollary 21, and Algorithm 3, are novel and have not been previously published. Also, the simulation scenarios are new and include a sensitivity analysis of RAM against various failure types (in [62], we tested RAM only against worst-case failures, and in different scenarios). Finally, all proofs in [62] were omitted and are now included here.

The rest of the article is organized as follows: Section II presents RAM, and quantifies its minimal running time. Section III presents RAM's suboptimality bounds. Section IV presents RAM's numerical evaluations. Section V concludes this article. All proofs are found in the Appendix.

II. Adaptive Algorithm: RAM

We present RAM, the first scalable algorithm for RSM, formulated in (2). RAM's pseudo-code is given in Algorithm 1. Below, we first give an intuitive description of RAM, and then a step-by-step description. Also, we quantify its running time. RAM's suboptimality bounds are given in Section III.

[3]Similarly to (1), RSM is generally inapproximable: no polynomial time algorithm guarantees a constant suboptimality bound across all problem instances. For example, it is inapproximable for fundamental applications in control and machine learning such as *sensor selection for optimal Kalman filtering* [20], and *feature selection for sparse model fitting* [19]. Thus, in this article, we focus our analysis in per-instance suboptimality bounds.

[4]A suboptimality bound is called *optimal* when it is the tightest achievable bound among all polynomial time algorithms, given a worst-case family of f.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Robust Adaptive Maximization (RAM).

Input: RAM receives the inputs:
Offline: integer T; function $f: 2^{\mathcal{V}_1} \cup \cdots \cup 2^{\mathcal{V}_T} \mapsto \mathbb{R}$ such that f is non-decreasing and $f(\emptyset) = 0$; integers α_t, β_t such that $0 \leq \beta_t \leq \alpha_t \leq |\mathcal{V}_t|$, for all $t = 1, \ldots, T$;
Online: at each $t = 2, \ldots, T$, observed removal $\mathcal{B}_{t-1}$ from RAM's output $\mathcal{A}_{t-1}$.
Output: At each step $t = 1, \ldots, T$, set $\mathcal{A}_t$.

1: **for all** $t = 1, \ldots, T$ **do**
2: $\mathcal{S}_{t,1} \leftarrow \emptyset$; $\mathcal{S}_{t,2} \leftarrow \emptyset$;
3: Sort elements in $\mathcal{V}_t$ s.t. $\mathcal{V}_t \equiv \{v_{t,1}, \ldots, v_{t,|\mathcal{V}_t|}\}$ and $f(v_{t,1}) \geq \cdots \geq f(v_{t,|\mathcal{V}_t|})$;
4: $\mathcal{S}_{t,1} \leftarrow \{v_{t,1}, \ldots, v_{t,\beta_t}\}$;
5: **while** $|\mathcal{S}_{t,2}| < \alpha_t - \beta_t$ **do**
6: $x \in \arg\max_{y \in \mathcal{V}_t \setminus (\mathcal{S}_{t,1} \cup \mathcal{S}_{t,2})} f(\mathcal{A}_1 \setminus \mathcal{B}_1, \ldots, \mathcal{A}_{t-1} \setminus \mathcal{B}_{t-1}, \mathcal{S}_{t,2} \cup \{y\})$;
7: $\mathcal{S}_{t,2} \leftarrow \{x\} \cup \mathcal{S}_{t,2}$;
8: **end while**
9: $\mathcal{A}_t \leftarrow \mathcal{S}_{t,1} \cup \mathcal{S}_{t,2}$.
10: **end for**

A. Intuitive Description

RSM aims to maximize f through a sequence of steps despite compromises to each step. Specifically, at each $t = 1, 2, \ldots$, RSM selects an $\mathcal{A}_t$ toward a maximal f despite the fact that $\mathcal{A}_t$ will be compromised by a worst-case removal $\mathcal{B}_t$, resulting to f being evaluated at $\mathcal{A}_1 \setminus \mathcal{B}_1, \ldots, \mathcal{A}_T \setminus \mathcal{B}_T$ instead of $\mathcal{A}_1, \ldots, \mathcal{A}_T$. In this context, RAM aims to achieve RSM's goal by selecting $\mathcal{A}_t$ as the union of two sets $\mathcal{S}_{t,1}$, and $\mathcal{S}_{t,2}$ (RAM's line 9), whose role we describe intuitively as follows.

$\mathcal{S}_{t,1}$ approximates (aims to guess the) worst-case removal from $\mathcal{A}_t$: With $\mathcal{S}_{t,1}$, RAM aims to capture the worst-case removal of β_t elements from $\mathcal{A}_t$. Intuitively, $\mathcal{S}_{t,1}$ is aimed to act as a "bait" to a worst-case attacker that selects the best β_t elements to remove from $\mathcal{A}_t$ at time t (*best* with respect to their contribution toward RSM's goal). RAM aims to approximate them by letting $\mathcal{S}_{t,1}$ be the set of β_t elements with the largest marginal contributions to f (RAM's lines 3–4). As such, each $\mathcal{S}_{t,1}$ is independent of the history of actual removals $\mathcal{B}_1, \ldots, \mathcal{B}_{t-1}$ and can be computed offline, before any of the $\mathcal{B}_1, \ldots, \mathcal{B}_T$ has been realized. In contrast, $\mathcal{S}_{t,2}$ can only be computed online, as we describe as follows.

$\mathcal{S}_{t,1} \cup \mathcal{S}_{t,2}$ approximates optimal solution to RSM's tth step: To complete $\mathcal{A}_t$s construction, RAM needs to select a set $\mathcal{S}_{t,2}$ of $\alpha_t - \beta_t$ elements (since $|\mathcal{A}_t| = \alpha_t$ and $|\mathcal{S}_{t,1}| = \beta_t$), and return $\mathcal{A}_t = \mathcal{S}_{t,1} \cup \mathcal{S}_{t,2}$ (RAM's line 9). Assuming $\mathcal{S}_{t,1}$s removal from $\mathcal{A}_t$, for $\mathcal{A}_t$ to be an optimal solution to RSM's tth maximization step, RAM needs to select $\mathcal{S}_{t,2}$ as a *best* set of $\alpha_t - \beta_t$ elements from $\mathcal{V}_t \setminus \mathcal{S}_{t,1}$. Nevertheless, this problem is NP-hard [17]. Thereby, RAM approximates such a best set, using the greedy procedure in RAM's lines 5–8. Particularly, RAM's line 6 adapts $\mathcal{S}_{t,2}$ to the history of removals $\mathcal{B}_1, \ldots, \mathcal{B}_{t-1}$ and selections $\mathcal{A}_1, \ldots, \mathcal{A}_{t-1}$, since it constructs $\mathcal{S}_{t,2}$ given $\mathcal{A}_1 \setminus \mathcal{B}_1, \ldots, \mathcal{A}_{t-1} \setminus \mathcal{B}_{t-1}$. As such, each $\mathcal{S}_{t,2}$, in contrast to $\mathcal{S}_{t,1}$, can be computed only online, only once the history of removals $\mathcal{B}_1, \ldots, \mathcal{B}_{t-1}$ has been realized.

Overall, RAM adaptively constructs an $\mathcal{A}_t$ to approximate an optimal solution to RSM's tth maximization step.

Remark 1 (Further intuition on why $\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$ are selected as in: RAM): We first discuss why RAM (i) selects $\mathcal{A}_t$ as the union of $\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$, and (ii) selects $\mathcal{S}_{t,2}$ as a greedily picked subset of $\mathcal{V}_t \setminus \mathcal{S}_{t,1}$. We then focus on $\mathcal{S}_{t,1}$.

If $\mathcal{S}_{t,1}$ guesses correctly the removal $\mathcal{B}_t$ from $\mathcal{A}_t = \mathcal{S}_{t,1} \cup \mathcal{S}_{t,2}$, then all elements in $\mathcal{S}_{t,2}$ remain intact *($\mathcal{A}_t \setminus \mathcal{B}_t = \mathcal{S}_{t,2}$)*. Therefore, since $\mathcal{S}_{t,2}$ has been selected using the greedy algorithm in RAM's lines 5–8, which is an optimal approximation algorithm for maximizing monotone functions subject to cardinality constraints [23],[5] $\mathcal{A}_t$ is an optimal approximation to RSM's tth maximization step. This explains why RAM selects $\mathcal{A}_t$ as the union of two sets ($\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$), and why RAM selects $\mathcal{S}_{t,2}$ greedily from $\mathcal{V}_t \setminus \mathcal{S}_{t,1}$ given $\mathcal{S}_{t,1}$. Generally, if $\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}$ guess $\mathcal{B}_1, \ldots, \mathcal{B}_T$ correctly, then RSM becomes equivalent to the attack-free (1), and RAM becomes equivalent to the optimal greedy algorithm for (1).

But if $\mathcal{S}_{t,1}$ guesses incorrectly $\mathcal{B}_t$, then some of $\mathcal{S}_{t,1}$s elements will survive, and, instead, some of $\mathcal{S}_{t,2}$s elements will be removed. The question arising now is: Can the elements that survived in $\mathcal{S}_{t,1}$ compensate for the removed elements from $\mathcal{S}_{t,2}$? In this article, we develop tools to prove that if the elements of $\mathcal{S}_{t,1}$ are chosen as in RAM's lines 3–4, this is indeed the case (proof of Theorem 10), providing the first provable approximation guarantees for RSM via RAM.

B. Step-By-Step Description

RAM executes four steps for each $t = 1, \ldots, T$: *Initialization (RAM's line 2):* RAM defines two auxiliary sets, namely, $\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$, and initializes them with the empty set (RAM's line 2).

Construction of set $\mathcal{S}_{t,1}$ (RAM's lines 3–4): RAM constructs $\mathcal{S}_{t,1}$ by selecting β_t elements, among all $s \in \mathcal{V}_t$, with the highest values $f(s)$. In detail, $\mathcal{S}_{t,1}$ is constructed by first indexing the elements in $\mathcal{V}_t$ such that $\mathcal{V}_t \equiv \{v_{t,1}, \ldots, v_{t,|\mathcal{V}_t|}\}$ and $f(v_{t,1}) \geq \cdots \geq f(v_{t,|\mathcal{V}_t|})$ (RAM's line 3), and then by including in $\mathcal{S}_{t,1}$ the first β_t elements (RAM's line 4).

Construction of set $\mathcal{S}_{t,2}$ (RAM's lines 5–8): RAM constructs $\mathcal{S}_{t,2}$ by picking greedily $\alpha_t - \beta_t$ elements from $\mathcal{V}_t \setminus \mathcal{S}_{t,1}$, taking also into account the history of selections and removals, that is, $\mathcal{A}_1 \setminus \mathcal{B}_1, \ldots, \mathcal{A}_{t-1} \setminus \mathcal{B}_{t-1}$. Specifically, the "while loop" (RAM's lines 5–8) selects an element $y \in \mathcal{V}_t \setminus (\mathcal{S}_{t,1} \cup \mathcal{S}_{t,2})$ to add in $\mathcal{S}_{t,2}$ only if y maximizes the value of $f(\mathcal{A}_1 \setminus \mathcal{B}_1, \ldots, \mathcal{A}_{t-1} \setminus \mathcal{B}_{t-1}, \mathcal{S}_{t,2} \cup \{y\})$.

Construction of set $\mathcal{A}_t$ (RAM's line 9): RAM constructs $\mathcal{A}_t$ as the union of $\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$.

The above-mentioned steps are valid for any number of failures β_t.

C. Running Time

We now analyze the computational complexity of RAM.

[5] An approximation algorithm is called *optimal* when it achieves the tightest possible achievable suboptimality bound among all polynomial time algorithms, given a worst-case family of functions f.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

Proposition 2: At each $t = 1, 2, \ldots$, RAM runs in $O[|\mathcal{V}_t|(\alpha_t - \beta_t)\tau_f]$ time, where τ_f is fs evaluation time.

Remark 3 (Minimal running time): Even though RAM robustifies the traditional, failure-free sequential optimization in (1), RAM has the same order of running time as the state-of-the-art algorithms for (1) [18, Sec. 4] [23, Sec. 8].

In summary, RAM selects adaptively a solution for RSM, in minimal running time, and is valid for any number of failures. We quantify its approximation performance as follows.

III. Suboptimality Guarantees

We present RAM's suboptimality bounds. We first present RAM's *a priori* bounds, and, then, the *a posteriori* bounds. Finally, we present the latter's prefailure approximations.

A. Curvature and Total Curvature

To present RAM's suboptimality bounds, we use the notions of *curvature* and *total curvature*. To this end, we start by recalling the definitions of *modularity* and *submodularity*, where we consider the notation.

1) $\mathcal{V} \triangleq \bigcup_{i=1}^{T} \mathcal{V}_t$, i.e., $\mathcal{V}$ is the union across the horizon T of all the available elements to choose from.

Definition 4 (Modularity [63]): $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is modular if and only if $f(\mathcal{A}) = \sum_{v \in \mathcal{A}} f(v)$, for any $\mathcal{A} \subseteq \mathcal{V}$.

Therefore, if f is modular, then $\mathcal{V}$s elements complement each other through f. Particularly, Definition 4 implies $f(\{v\} \cup \mathcal{A}) - f(\mathcal{A}) = f(v)$, for any $\mathcal{A} \subseteq \mathcal{V}$ and $v \in \mathcal{V} \setminus \mathcal{A}$.

Definition 5 (Submodularity [63]): $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is *submodular* if and only if $f(\mathcal{A} \cup \{v\}) - f(\mathcal{A}) \geq f(\mathcal{B} \cup \{v\}) - f(\mathcal{B})$, for any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $v \in \mathcal{V}$.

The definition implies f is submodular if and only if the return $f(\mathcal{A} \cup \{v\}) - f(\mathcal{A})$ diminishes as $\mathcal{A}$ grows, for any v. In contrast to f being modular, if f is submodular, then $\mathcal{V}$s elements substitute each other. Specifically, without loss of generality, consider f to be nonnegative: then, Definition 5 implies $f(\{v\} \cup \mathcal{A}) - f(\mathcal{A}) \leq f(v)$. That is, in the presence of $\mathcal{A}$, vs contribution to $f(\{v\} \cup \mathcal{A})$s value is diminished.

Definition 6: (Curvature [27]). Consider a nondecreasing submodular $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $f(v) \neq 0$, for any $v \in \mathcal{V}$, without loss of generality. Then, *f's curvature* is defined as

$$\kappa_f \triangleq 1 - \min_{v \in \mathcal{V}} \frac{f(\mathcal{V}) - f(\mathcal{V} \setminus \{v\})}{f(v)}. \tag{3}$$

Definition 6 implies $\kappa_f \in [0, 1]$. Particularly, κ_f measures how far f is from modularity: if $\kappa_f = 0$, then $f(\mathcal{V}) - f(\mathcal{V} \setminus \{v\}) = f(v)$, for all $v \in \mathcal{V}$; that is, f is modular. In contrast, if $\kappa_f = 1$, then there exist $v \in \mathcal{V}$ such that $f(\mathcal{V}) = f(\mathcal{V} \setminus \{v\})$; that is, v has no contribution to $f(\mathcal{V})$ in the presence of $\mathcal{V} \setminus \{v\}$. Therefore, κ_f can also been interpreted as a measure of how much $\mathcal{V}$s elements complement/substitute each other.

Definition 7 (Total curvature [23], [30]): Consider a monotone $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$. Then, fs total curvature is defined as

$$c_f \triangleq 1 - \min_{v \in \mathcal{V}} \min_{\mathcal{A}, \mathcal{B} \subseteq \mathcal{V} \setminus \{v\}} \frac{f(\{v\} \cup \mathcal{A}) - f(\mathcal{A})}{f(\{v\} \cup \mathcal{B}) - f(\mathcal{B})}. \tag{4}$$

Algorithm 2: Online Greedy Algorithm [18, Section 4].

Input: Integer $T > 0$; $f : 2^{\mathcal{K}_1} \cup \cdots \cup 2^{\mathcal{K}_T} \mapsto \mathbb{R}$ such that f is non-decreasing and $f(\emptyset) = 0$; integers δ_t such that $0 \leq \delta_t \leq |\mathcal{K}_t|$, for all $t = 1, \ldots, T$.
Output: At each step $t = 1, \ldots, T$, set $\mathcal{M}_t$.

```
for all t = 1, ..., T do
  M_t ← ∅;
  while |M_t| < δ_t do
    x ∈ arg max_{y ∈ K_t \ M_t} f(M_1, ..., M_{t-1}, M_t ∪ {y});
    M_t ← {x} ∪ M_t;
  end while
end for
```

Similarly to κ_f, it also is $c_f \in [0, 1]$. Remarkably, when f is submodular, then $c_f = \kappa_f$. Generally, if $c_f = 0$, then f is modular, while if $c_f = 1$, then (4) implies the assumption that f is nondecreasing. In [64], any monotone f with total curvature c_f is called c_f-submodular, as repeated as follows.[6]

Definition 8 (c_f-submodularity [64]): Any monotone function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ with total curvature c_f is called *c_f-submodular*.

Remark 9 (Dependence on the size of $\mathcal{V}$ and length of horizon T): Evidently, both κ_f and c_f are nondecreasing as $\mathcal{V}$ grows (cf. Definitions 6 and 7). Therefore, κ_f and c_f are also nondecreasing as T increases, since $\mathcal{V} \equiv \bigcup_{i=1}^{T} \mathcal{V}_t$.

B. Priori Suboptimality Bounds

We present RAM's *a priori* suboptimality bounds, using the abovementioned notions of curvature. We use also the notation as follows:

1) $f^\star$ is the optimal value of RSM;
2) $\mathcal{A}_{1:t} \triangleq (\mathcal{A}_1, \ldots, \mathcal{A}_t)$, where $\mathcal{A}_t$ is the selected set by RAM at $t = 1, 2, \ldots$;
3) $(\mathcal{B}_1^\star, \ldots, \mathcal{B}_T^\star)$ is an optimal removal from $\mathcal{A}_{1:T}$;
4) $\mathcal{B}_{1:t}^\star \triangleq (\mathcal{B}_1^\star, \ldots, \mathcal{B}_t^\star)$;
5) $\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star \triangleq (\mathcal{A}_1 \setminus \mathcal{B}_1^\star, \ldots, \mathcal{A}_t \setminus \mathcal{B}_t^\star)$.

Theorem 10 (: A priori **bounds)** RAM selects $\mathcal{A}_{1:T}$ such that $|\mathcal{A}_t| \leq \alpha_t$, and if f is submodular, then

$$\frac{f(\mathcal{A}_{1:T} \setminus \mathcal{B}_{1:T}^\star)}{f^\star} \geq \begin{cases} \frac{1 - e^{-\kappa_f}}{\kappa_f}(1 - \kappa_f), & T = 1 \\ (1 - \kappa_f)^4, & T > 1 \end{cases} \tag{5}$$

whereas, if f is c_f-submodular, then

$$\frac{f(\mathcal{A}_{1:T} \setminus \mathcal{B}_{1:T}^\star)}{f^\star} \geq \begin{cases} (1 - c_f)^3, & T = 1 \\ (1 - c_f)^5, & T > 1. \end{cases} \tag{6}$$

Evidently, Theorem 10 bounds are *a priori*, since right-hand sides of (5) and (6) are independent of the selected $\mathcal{A}_{1:T}$ by RAM, and the incurred failures $\mathcal{B}_{1:T}^\star$.

Importantly, the bounds compare RAM's selection $\mathcal{A}_{1:T}$ against an optimal one that knows *a priori* all future failures (and achieves that way the value $f^\star$). Instead, RAM's has no

[6] Lehmann *et al.* [64] defined c_f-submodularity by considering in (4) $\mathcal{A} \subseteq \mathcal{B}$ instead of $\mathcal{A} \subseteq \mathcal{V}$. Generally, nonsubmodular but monotone functions have been referred to as *approximately* or *weakly* submodular [29], [65], names that have also been adopted for the definition of c_f in [64], e.g., in [66] and [67].

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

knowledge of the future failures. Within this challenging setting, Theorem 10 nonetheless implies: for functions f with $\kappa_f < 1$ or $c_f < 1$, RAM's selection $\mathcal{A}_{1:T}$ is finitely close to the optimal, instead of arbitrarily suboptimal. Indeed, then Theorem 10 bounds are nonzero. We discuss functions with $\kappa_f < 1$ or $c_f < 1$ below, along with relevant applications.

Remark 11 (Functions with $\kappa_f < 1, c_f < 1$, and applications): Functions with $\kappa_f < 1$ are the concave over modular functions [28, Sec. 2.1] and the $\log\det$ of positive-definite matrices [68]. Also, functions with $c_f < 1$ are the support selection functions [65], the average minimum square error of the Kalman filter (trace of error covariance) [69, Sec. IV], and the linear-quadratic-Gaussian (LQG) cost as a function of the active sensors [10, Th. 4]. The aforementioned functions appear in control and machine learning applications such as feature selection [21], [70], and actuator and sensor scheduling [5]–[13], [69].

Evidently, when κ_f and c_f tend to 0, then RAM becomes optimal, since all bounds in Theorem 10 tend to 1; for example, $1/\kappa_f(1-e^{-\kappa_f})(1-\kappa_f)$ increases as κ_f decreases, and its limit is equal to 1 for $\kappa_f \to 0$. Application examples of this sort involve the regression of Gaussian processes with RBF kernels [68, Th. 5], such as in sensor selection for temperature monitoring [71].

Finally, since both κ_f and c_f are nondecreasing in T and $\mathcal{V}$ (Remark 9), the bounds are nonincreasing in T and $\mathcal{V}$.

Tightness and optimality (toward a posteriori bounds). RAM's curvature-dependent bounds are the first suboptimality bounds for RSM, and make a first step toward separating the classes of monotone functions into functions for which RSM can be approximated well (low curvature functions), and functions for which it cannot (high curvature functions). Moreover, although for the failure-free (1) the *a priori* bounds $1/\kappa_f(1-e^{-\kappa_f})$ and $1/(1+\kappa_f)$ (where f is submodular) are known to be tight [27, Ths. 2.12 and 5.4], the tightness of (5) is an open problem. Similarly, although for (1), the *a priori* bound $1-c_f$ (where f is c_f-submodular) is known to be optimal (the tightest possible in polynomial time in a worst-case) [23, Th. 8.6], the optimality of (6) is an open problem. Notably, in the latter case (f is c_f-submodular) both $1-c_f$ and the bound in (6) are 0 for $c_f = 1$, which is in agreement with the inapproximability of both (1) and RSM in the worst-case.

In contrast to Theorem 10 *a priori* bounds, we next present tight and/or optimal *a posteriori* bounds.

C. A Posteriori Suboptimality Bounds

We now present RAM's *a posteriori* bounds, which are computable once all failures up to step t have been observed. Henceforth, we use the notation as follows:

1) $f^\star_t$ is the optimal value of RSM for $T = t$;
2) $\mathcal{M}_t$ is the set returned by the online, failure-free greedy Algorithm 2 at $t = 1, \ldots, T$,[7] when we consider therein $\delta_t = \alpha_t - \beta_t$ and $\mathcal{K}_t = \mathcal{V}_t \setminus \mathcal{S}_{1,t}$;
3) $\mathcal{M}_{1:t} \triangleq \{\mathcal{M}_1, \ldots, \mathcal{M}_t\}$.

Remark 12 (Interpretation of $\mathcal{M}_{1:t}$): Since each $\mathcal{S}_{1,t}$ is the expected future failures ("baits") selected in RAM's lines 3–4 (see Section II), $\mathcal{M}_{1:t}$ are the sets one would greedily select per Algorithm 2 if it was known a priori that indeed the future failures are the $\mathcal{S}_{1,t}$, $t = 1, \ldots, T$.

Theorem 13 (: A posteriori bounds) For all $t = 1, \ldots, T$, given the observed history $\mathcal{B}^\star_{1:t}$, RAM selects $\mathcal{A}_t$ such that $|\mathcal{A}_t| \leq \alpha_t$, and if f is submodular, then

$$\frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{f^\star_t} \geq \begin{cases} \frac{1-e^{-\kappa_f}}{\kappa_f} \frac{f(\mathcal{A}_1 \setminus \mathcal{B}^\star_1)}{f(\mathcal{M}_1)}, & t = 1 \\ \frac{1}{1+\kappa_f} \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{f(\mathcal{M}_{1:t})}, & t > 1 \end{cases} \tag{7}$$

whereas, if f is c_f-submodular, then

$$\frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{f^\star_t} \geq (1-c_f) \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{f(\mathcal{M}_{1:t})}. \tag{8}$$

Theorem 14 (Tightness and optimality): There exist families of f such that the suboptimality bounds in (7) are tight. Also, there exist families of f such that the suboptimality bounds in (8) are optimal (the tightest possible) across all algorithms that evaluate f a polynomial number of times.[8]

The bounds break down into the *a priori* κ_f- and c_f-depended parts, and the *a posteriori* $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t})$. We refer to the latter as *a posteriori* since it is computable *after* $\mathcal{B}^\star_t$ has been observed. Intuitively, the *a posteriori* part captures how successful the "bait" $\mathcal{S}_{1,t}$ has been in approximating the anticipated worst-case failure $\mathcal{B}^\star_t$. Indeed, if $\mathcal{B}^\star_t = \mathcal{S}_{1,t}$ for all $t = 1, 2, \ldots$, then $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t}) = 1$ and Theorem 13 bounds become the tight/optimal *a priori* bounds $1/\kappa_f(1-e^{-\kappa_f})$, $1/(1+\kappa_f)$, and $1-c_f$,[9] and, as such, they are also tighter than Theorem 10 *a priori* bounds.

In general, Theorem 13 *a posteriori* bounds may be looser than Theorem 10 *a priori* bounds; yet, they are tighter when $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t})$ is close enough to 1, e.g., for f being c_f-submodular and $T > 1$, if $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t}) > (1-c_f)^4$, then the *a posteriori* bound in (8) is tighter than the *a priori* in (6). Indeed, in the numerical evaluations of Section IV, $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t})$ is nearly 1, whereas $(1-c_f)^4 \leq .0001$; thus, (8) is three orders tighter than (6).

Notably, the *a priori* parts $1/\kappa_f(1-e^{-\kappa_f})$, $1/(1+\kappa_f)$ are nonzero for any values of κ_f. In more detail, $1/\kappa_f(1-e^{-\kappa_f}) \geq 1-1/e$ and $1/(1+\kappa_f) \geq 1/2$ for all $\kappa_f \in [0,1]$; particularly, $1/\kappa_f(1-e^{-\kappa_f})$ increases as κ_f decreases, and its limit is equal to 1 for $\kappa_f \to 0$. Therefore, in contrast to the *a priori* bound in (5), which for $\kappa_f = 1$ becomes 0, (7) for $\kappa_f = 1$ becomes instead

$$\frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{f^\star_t} \geq \begin{cases} (1-1/e)\frac{f(\mathcal{A}_1 \setminus \mathcal{B}^\star_1)}{f(\mathcal{M}_1)}, & t = 1 \\ \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})}{2f(\mathcal{M}_{1:t})}, & t > 1. \end{cases} \tag{9}$$

Nevertheless, such simplification for (8) is not evident, a fact that is in agreement with both (i) RSM's inapproximability when

[7] We refer to Algorithm 2 as "online" since each $\mathcal{M}_t$ can be chosen in real time (at time t) sequentially, i.e., given the history of past selections $\mathcal{M}_1, \ldots, \mathcal{M}_{t-1}$. Observe, however, that if one wishes so one can also execute all steps of Algorithm 2 offline at time $t = 0$.

[8] Theorem 14 function families are the same as those in the proofs of [27, Ths. 2.12 and 5.4] and [23, Th. 8.6], which prove the tightness and optimality of $1/\kappa_f(1-e^{-\kappa_f})$, $1/(1+\kappa_f)$, and $(1-c_f)$ for (1).

[9] Theorem 14 is proved based on this observation.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

f is *not* submodular, necessitating per-instance suboptimality bounds for any polynomial time algorithm, and (ii) (8) optimality per Theorem 14.

Overall, Theorem 13 bounds are computable online, at each $t = 1, 2, \ldots$, *after* failure $\mathcal{B}_t^\star$ has been observed. We next approximate the bounds *before* $\mathcal{B}_t^\star$ occurs.

D. Prefailure Approximations of Postfailure Bounds

We present prefailure approximations to Theorem 13 postfailure bounds. In particular, we propose a method to lower bound $f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ by a value $\hat{f}_t$, at each $t = 1, \ldots, T$, given $\mathcal{B}_{1:t-1}^\star$ (but before $\mathcal{B}_t^\star$ occurs).

In more detail, we recall $f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ is the value of the constrained optimization problem

$$f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star) \equiv \min_{\mathcal{B}_t \subseteq \mathcal{A}_t,\, |\mathcal{B}_t| \leq \beta_t} f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}^\star, \mathcal{A}_t \setminus \mathcal{B}_t). \tag{10}$$

Computing $f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ is NP-hard, even if f is submodular [72]. But lower bounding $f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ can be efficient. Specifically, the nonconstrained reformulation of (10) in (11) below is efficiently solvable (see [72]–[75] for f being submodular; and [76] for f being c_f-submodular)

$$\min_{\mathcal{B}_t \subseteq \mathcal{A}_t} f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}^\star, \mathcal{A}_t \setminus \mathcal{B}_t) + \lambda_t |\mathcal{B}_t| \tag{11}$$

where $\lambda_t \geq 0$ and constant (λ_t acts similarly to a Lagrange multiplier [77]). Evidently, Lemma 15 holds true, where $\hat{\mathcal{B}}_t(\lambda_t)$ denotes an optimal solution to (11), i.e.,

$$\hat{\mathcal{B}}_t(\lambda_t) \in \arg\min_{\mathcal{B}_t \subseteq \mathcal{A}_t} f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}^\star, \mathcal{A}_t \setminus \mathcal{B}_t) + \lambda_t |\mathcal{B}_t| \tag{12}$$

and where $\hat{f}_t(\lambda_t)$ denotes the value of $f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}^\star, \mathcal{A}_t \setminus \mathcal{B}_t)$ when $\mathcal{B}_t = \hat{\mathcal{B}}_t(\lambda_t)$, i.e.,

$$\hat{f}_t(\lambda_t) \triangleq f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}^\star, \mathcal{A}_t \setminus \hat{\mathcal{B}}_t(\lambda_t)). \tag{13}$$

Lemma 15: There exists $\lambda_t^\star$ such that $\hat{f}_t(\lambda_t) \leq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ and $|\hat{\mathcal{B}}_t(\lambda_t)| > \beta_t$ for $\lambda_t < \lambda_t^\star$; whereas, $\hat{f}_t(\lambda_t) \geq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$ and $|\hat{\mathcal{B}}_t(\lambda_t)| \leq \beta_t$ for $\lambda_t \geq \lambda_t^\star$.

To observe such a value $\lambda_t^\star$ exists, it suffices to observe: (i) for $\lambda_t = 0$, the cardinality of $\hat{\mathcal{B}}_t$ in (11) is unconstrained, and, thus, the optimal solution in (11) is to remove all $\mathcal{A}_t$, which implies $|\hat{\mathcal{B}}_t(\lambda_t)| = \alpha_t \geq \beta_t$; (ii) more generally, for $\lambda_t > 0$, the cardinality of $\hat{\mathcal{B}}_t$ in (11) is increasingly penalized as λ_t increases, and, thus, $|\hat{\mathcal{B}}_t(\lambda_t)|$ is a decreasing function of λ_t (in particular, if $\lambda_t \to +\infty$, then $|\hat{\mathcal{B}}_t(\lambda_t)| \to 0$). Now, given (i) and (ii), denote by $\lambda_t^\star$ the first value of λ_t such that $|\hat{\mathcal{B}}_t(\lambda_t)| \leq \beta_t$, when λ_t is initially set to 0 and then continuously increases: then, for $\lambda_t < \lambda_t^\star$, it is $|\hat{\mathcal{B}}_t(\lambda_t)| > \beta_t$, and, thus, $\hat{f}_t(\lambda_t) \leq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$, since $|\hat{\mathcal{B}}_t(\lambda_t)| > \beta_t = |\mathcal{B}_t^\star|$ and $\hat{\mathcal{B}}_t(\lambda_t)$ is an optimal solution to (11); whereas, for $\lambda_t \geq \lambda_t^\star$, it is $|\hat{\mathcal{B}}_t(\lambda_t)| \leq \beta_t$, and, thus, $\hat{f}_t(\lambda_t) \geq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$.

Although $\lambda_t^\star$ is unknown, it can be approximated by using bisection. For example, Algorithm 3 uses bisection with accuracy level $\epsilon > 0$ (Algorithm 3 lines 2–10) to find a λ_t that is ϵ-close to $\lambda_t^\star$ and for which $\hat{f}_t(\lambda_t) \leq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$. To start the bisection, Algorithm 3 assumes a large enough $u_0 \geq 0$ such that $|\hat{\mathcal{B}}_t(u_0)| < \beta_t$; such a u_0 can be found since $|\hat{\mathcal{B}}_t(u_0)| \to 0$ for $u_0 \to +\infty$. Next, at each "while loop" (lines 2–10 of Algorithm 3), $\lambda_t^\star \in [l, u]$, since $|\hat{\mathcal{B}}_t(u)| < \beta_t$ and $|\hat{\mathcal{B}}_t(l)| \geq \beta_t$ (cf. line 5 and line 7 of Algorithm 3). Per line 2 of the algorithm, l and u are updated until $u - l \leq \epsilon$. Then, after at most $\log_2[(u-l)/\epsilon]$ iterations, the algorithm terminates by setting λ_t equal to the latest value of l (lines 11–12 of the algorithm). Therefore, λ_t is ϵ-close to $\lambda_t^\star$ and satisfies $|\hat{\mathcal{B}}_t(l)| \geq \beta_t$, which in turn implies $\hat{f}_t(\lambda_t) \leq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$, as desired. All in all, given an approximation λ_t to $\lambda_t^\star$, Lemma 15 implies the following approximation of Theorem 13 bounds.

Algorithm 3: Bisection.

Input: Integer β_t per RSM; function f per RSM; histories $\mathcal{A}_{1:t}$ and $\mathcal{B}_{1:t-1}^\star$; $u_0 > 0$ such that $|\hat{\mathcal{B}}_t(u_0)| < \beta_t$, where $\hat{\mathcal{B}}_t(\cdot)$ is defined in eq. (12); $\epsilon > 0$, which defines bise- ction's stopping condition (accuracy level).

Output: $\lambda_t \geq 0$ such that $\hat{f}_t(\lambda_t) \leq f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)$, where λ_t and $\hat{f}_t(\lambda_t)$ are defined in eq. (11) and eq. (13).

1: $l \leftarrow 0$; $u \leftarrow u_0$; $\lambda_t \leftarrow (l+u)/2$;
2: **while** $u - l > \epsilon$ **do**
3: Find $\hat{\mathcal{B}}_t(\lambda_t)$ by solving eq. (12);
4: **if** $|\hat{\mathcal{B}}_t(\lambda_t)| < \beta_t$ **then**
5: $u \leftarrow \lambda_t$; ▷u always satisfies $|\hat{\mathcal{B}}_t(u)| < \beta_t$
6: **else**
7: $l \leftarrow \lambda_t$; ▷l always satisfies $|\hat{\mathcal{B}}_t(l)| \geq \beta_t$
8: **end if**
9: $\lambda_t \leftarrow (l+u)/2$;
10: **end while**
11: $\lambda_t \leftarrow l$; ▷l is ϵ-close to $\lambda_t^\star$ ($\lambda_t^\star$ is defined in Lemma 15) and satisfies $|\hat{\mathcal{B}}_t(l)| \geq \beta_t$
12: **return** λ_t.

Corollary 16 (Prefailure approximation of a posteriori bounds): Let Algorithm 3 return λ_t, for $t = 1, \ldots, T$. RAM selects $\mathcal{A}_t$ such that $|\mathcal{A}_t| \leq \alpha_t$, and if f is submodular, then

$$\frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)}{f_t^\star} \geq \begin{cases} \frac{1-e^{-\kappa_f}}{\kappa_f} \frac{\hat{f}_1(\lambda_1)}{f(\mathcal{M}_1)}, & t = 1 \\ \frac{1}{1+\kappa_f} \frac{\hat{f}_t(\lambda_t)}{f(\mathcal{M}_{1:t})}, & t > 1 \end{cases} \tag{14}$$

whereas if f is c_f-submodular, then

$$\frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^\star)}{f_t^\star} \geq (1 - c_f) \frac{\hat{f}_t(\lambda_t)}{f(\mathcal{M}_{1:t})}. \tag{15}$$

Corollary 16 describes an online mechanism to predict the RAM's performance before the upcoming failures, step by step.

Remark 17 (Utility of Corollary 16 bounds): For $T = 1$, Corollary 16 bounds are computed before $\mathcal{B}_1^\star$ occurs, which implies $\hat{f}_1$ can be computed *offline (*at any time $t < 1$*)*. Therefore, Corollary 16 bounds, when tighter than Theorem 10 a priori bounds, allow for an a priori assessment of RAM's approximation performance *(before* RAM is deployed in the real world*)*. Examples of 1-step design problems where $T = 1$, include *robust actuator and sensor placement* [7], [16], [78], *robust*

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

feature selection [21], [32], [79], *robust graph covering* [80], and *robust server placement* [15], [81].

For $T > 1$, Corollary 16 bounds can only be computed *online*, once $\mathcal{B}^\star_{1:t-1}$ has been observed (but before $\mathcal{B}^\star_t$ has occurred), for each $t = 1, \ldots, T$.[10] As such, the bounds can be used to balance the tradeoff between (i) computation time requirements (including computation and energy consumption requirements) for solving each step t of RSM, and (ii) approximation performance requirements for solving RSM at each t. For example, if Corollary 16 bounds indicate a good performance by RAM at t (e.g., the bounds are above a given threshold), RAM is used to select $\mathcal{A}_t$, since RAM is computation time inexpensive, being a polynomial time algorithm. However, if the bounds indicate poor performance by RAM at t (less than the given threshold), then an optimal algorithm can be used instead at t (such as the one proposed in [34]), but at the expense of higher computation time, since any optimal algorithm is nonpolynomial in the worst-case.[11]

IV. Applications

We evaluate the RAM's performance in applications. We start by assessing its near-optimality against worst-case failures. We continue by testing its sensitivity against non-worst-case failures, particularly, random and greedily selected failures. For such failures, one would expect RAM's performance to be the same, or improve, since RAM is designed to withstand the worst-case. To these ends, we consider two applications from the introduction: *sensor scheduling for autonomous navigation*, and *target tracking with WSNs*.

A. Sensor Scheduling for Autonomous Navigation

We demonstrate the RAM's performance in autonomous navigation scenarios, in the presence of sensing failures. We focus on small-scale instances, to enable RAM's comparison with a brute-force algorithm attaining the optimal to RSM. Instead, in Section IV-B, we consider larger-scale instances.

A UAV moves in a 3-D space, starting from a randomly selected initial location. Its objective is to land at $[0,\ 0,\ 0]$ with zero velocity. The UAV is modeled as a double-integrator with state $x_t = [p_t\ v_t]^\top \in \mathbb{R}^6$, where $t = 1, 2, \ldots$ is the time index, p_t is the UAV's position, and v_t is its velocity. The UAV controls its acceleration. The process noise has covariance $\mathbf{I}_6$.

The UAV is equipped with two on-board sensors: a GPS, measuring the UAV's position p_t with a covariance $2 \cdot \mathbf{I}_3$, and an altimeter, measuring p_ts altitude component with standard deviation 0.5 m. Also, the UAV can communicate with ten linear ground sensors. These sensors are randomly generated at each Monte Carlo run, along with their noise covariance.

The UAV has limited on-board battery power and measurement-processing bandwidth. Hence, it uses only a few sensors at each t. Particularly, among the 12 available sensors, the UAV uses at most α, where α varies from 1 to 12 in the Monte Carlo analysis (per RSM's notation, $\alpha_t = \alpha$ for all $t = 1, 2, \ldots$). The UAV selects the sensors to minimize the cumulative batch-state error over a horizon $T = 5$, captured by

$$c(\mathcal{A}_{1:t}) = \log\det[\Sigma_{1:t}(\mathcal{A}_{1:t})] \tag{16}$$

where $\Sigma_{1:t}(\mathcal{A}_{1:t})$ is the error covariance of the minimum variance estimator of $(x_1, \ldots, x_t)$ given the used sensors up to t [82]. Notably, $f(\mathcal{A}_{1:t}) = -c(\mathcal{A}_{1:t})$ is nondecreasing and submodular, in congruence to RSM's framework [8].

Finally, we consider that at most β failures are possible at each t (per RSM's notation, $\beta_t = \beta$ for all t). In the Monte Carlo analysis, β varies from 0 to $\alpha - 1$.

Baseline algorithms: We compare RAM with three algorithms. The first algorithm is a brute-force, optimal algorithm, denoted as `optimal`. Evidently, `optimal` is viable only for small-scale problem instances, such as herein where the available sensors are 12. The second algorithm performs random selection and is denoted as `random`. The third algorithm, denoted as `greedy`, greedily selects sensors to optimize (16) per the failure-free optimization setup in (1).

Results: The results are averaged over 100 Monte Carlo runs. For $\alpha = 8$ and $\beta = 4, 5, 6, 7$, they are reported in Fig. 1. For the remaining α and β values, the qualitative results are the same. From Fig. 1, the following observations are due.

Near-optimality against worst-case failures: We focus on first row of Fig. 1, where β varies from 4 to 7 (from left to right). Across all β, RAM nearly matches `optimal`. In contrast, `greedy` nearly matches `optimal` only for $\beta = 4$ (and, generally, for $\beta \leq \alpha/2$, taking into account the simulation results for the remaining values of α). Expectedly, `random` is always the worst among all compared algorithms. Importantly, as β tends to α, the `greedy`'s performance tends to `random`'s. The observation exemplifies the insufficiency of the traditional optimization paradigm in (1) against failures.

Across all values of α and β in the Monte Carlo analysis, the suboptimality bound in Theorem 13's (7) is at least 0.59, informing RAM performs at least 50% the optimal (κ_f remains always less than 0.93, while $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t})$ is close to 0.95). In contrast, in Fig. 1, we observe an almost optimal performance. This is an example where the actual performance of the algorithm is significantly closer to the optimal than what is indicated by the algorithm's suboptimality bound. Indeed, this is a common observation for greedy-like algorithms: for the failure-agnostic greedy in [18] see, e.g., [14].

Robustness against non-worst-case failures: We compare subfigures of Fig. 1 column-wise, where the failure type varies among worst-case, greedy, and random (from top to bottom).[12] Particularly, the RAM's performance remains the same, or improves, against non-worst-case failures, and the best performance is being observed against random failures, as expected. For example, if we focus on the rightmost column (where

[10] Corollary 16 bounds can only be computed *online* since RAM itself is an online algorithm, computing $\mathcal{A}_t$ only once $\mathcal{B}^\star_{1:t-1}$ has been observed (yet before $\mathcal{B}^\star_t$ has occurred), for each $t = 1, \ldots, T$.

[11] Even when RAM is used in combination with another algorithm to choose $\mathcal{A}_{1:t}$, Corollary 16 bounds are still applicable since they are *algorithm agnostic* (cf. proof of Theorem 13).

[12] We refer to a failure $\mathcal{B}_t$ as "greedy," when $\mathcal{B}_t$ is selected greedily toward minimizing $f(\mathcal{A}_{1:t-1} \setminus \mathcal{B}_{1:t-1}, \mathcal{A}_t \setminus \mathcal{B}_t)$, where $\mathcal{A}_{1:t}$ and $\mathcal{B}_{1:t-1}$ are given, as in Algorithm 2 but now for minimization instead of maximization.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

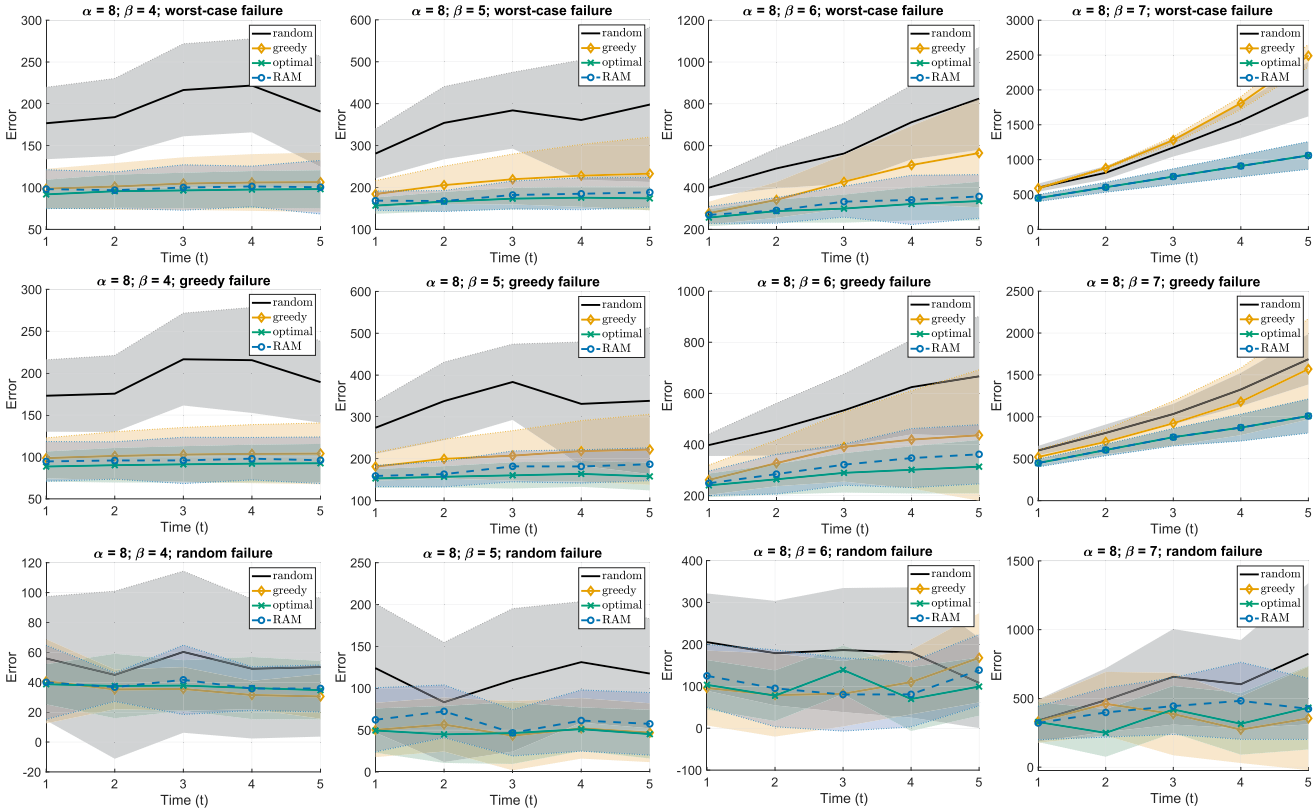


Fig. 1. Representative simulation results for the application *sensor scheduling for autonomous navigation*. Results are averaged across 100 Monte Carlo runs. Depicted is the estimation error for increasing time t, per (16), where $\alpha_t = \alpha = 8$ across all subfigures, whereas $\beta_t = \beta$ where β varies across subfigures column-wise. Finally, the failure type also varies, but row-wise. *Each subfigure has different scale.* .

$\alpha = 8; \beta = 7$), at $t = 5$, then we observe: for worst-case failures, RAM achieves error 1061; instead, for greedy failures, RAM achieves the reduced error 1010; while for random failures, RAM achieves even less error (less than 500). Finally, against greedy failures, RAM is still superior to `greedy`, while against random failures, they fare similarly.

Overall, the abovementioned numerical simulations demonstrate both the necessity for failure-robust optimization (RSM), as well as the near-optimality of RAM, even for increasing number of failures (system-wide failures). Similar conclusions we make over the second application scenario as follows.

B. Target Tracking With WSNs

We demonstrate the RAM's performance in adversarial target tracking scenarios. Particularly, we consider a mobile target who aims to escape detection from a WSN. To this end, the agent causes failures to the network.

A UAV (the target) is moving in a 3-D, cubic shaped space. The UAV moves on a straight line, across two opposite boundaries of the cube, keeping constant altitude and speed. The line's start and end points are randomly generated at each Monte Carlo run. The UAV's model is as in the autonomous navigation scenario in Section IV-A.

The WSN is composed of 100 ground sensors. It is aware of the UAV's model, but can only noisily observe its state. The sensors are randomly generated at each Monte Carlo run.

Due to power consumption and bandwidth limitations, only a few sensors can be active at each $t = 1, 2, \ldots$. Particularly, we assume $\alpha = 10$ active sensors at each t. Also, we assume the sensors are activated so the cumulative Kalman filtering error over a horizon $T = 5$ is minimized, as prescribed by

$$c(\mathcal{A}_{1:t}) = \sum_{t=1}^{T} \text{trace}[\Sigma_{t|t}(\mathcal{A}_{1:t})] \tag{17}$$

where $\Sigma_{t|t}(\mathcal{A}_{1:t})$ is the Kalman filtering error covariance. Noticeably, $f(\mathcal{A}_{1:t}) = -c(\mathcal{A}_{1:t})$ is nondecreasing and c_f-submodular, in agreement with RSM's framework [69].

Finally, at most β failures are possible at each t. In the Monte Carlo analysis, β varies from 1 to $\alpha - 1 = 9$.

Baseline algorithms: We compare RAM with `random`, and `greedy`. We cannot compare with `optimal`, since the network's large-scale size makes `optimal` unfeasible.

Results: The simulation results are averaged over 100 Monte Carlo runs. For $\beta = 3, 5, 7, 9$, they are reported in Fig. 2, where `random` is excluded since it results to exceedingly larger errors. For the remaining β values, the qualitative results remain the same. From Fig. 2, we make the observations.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

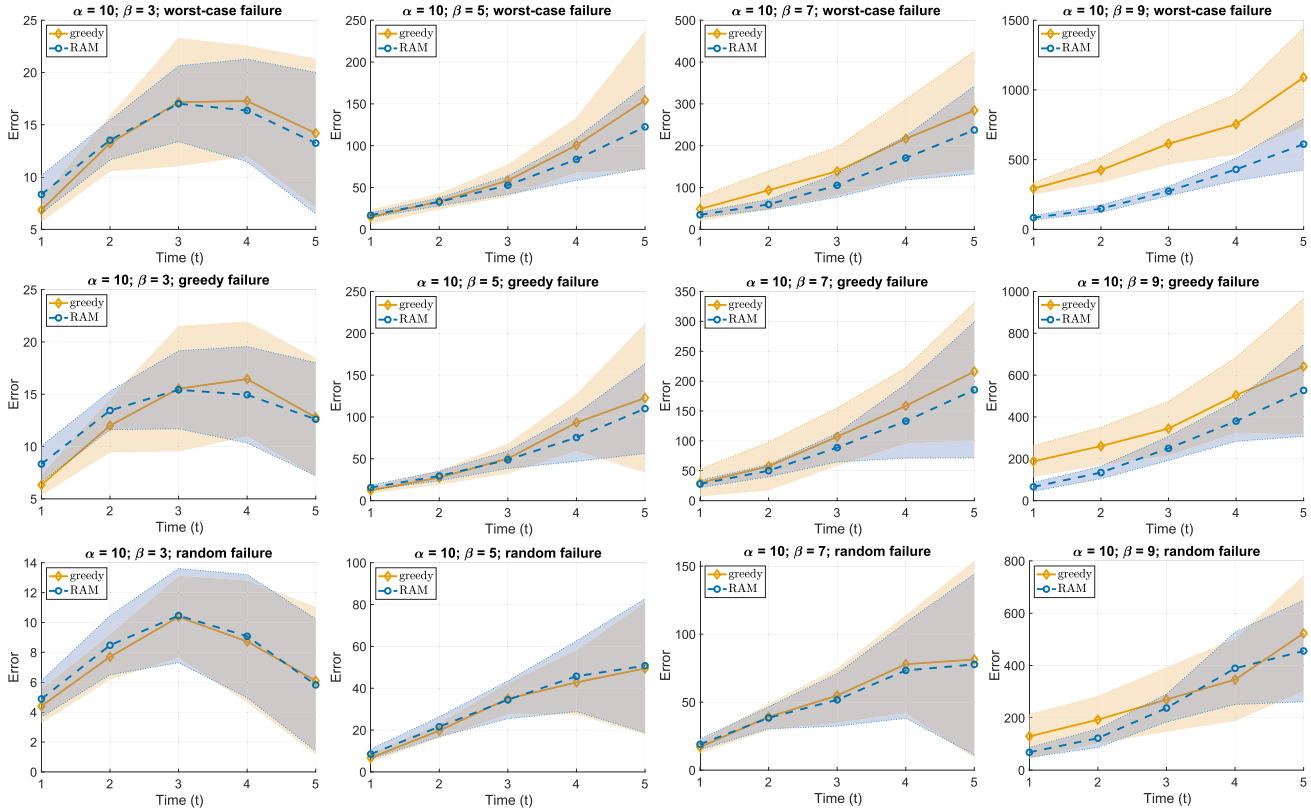


Fig. 2. Representative simulation results for the application target tracking with WSNs. Results are averaged over 100 Monte Carlo runs. Depicted is the estimation error for increasing t, per (17), where $\alpha_t = \alpha = 10$ across all subfigures, whereas $\beta_t = \beta$ where β varies across subfigures column-wise. Finally, the failure type also varies, but row-wise. Each subfigure has different scale.

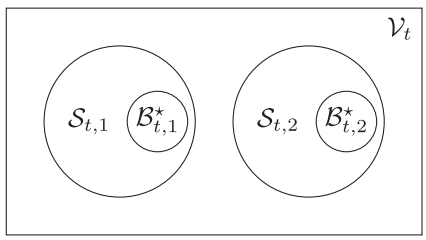


Fig. 3. Venn diagram, where the sets $\mathcal{S}_{t,1}, \mathcal{S}_{t,2}, \mathcal{B}^\star_{t,1}, \mathcal{B}^\star_{t,2}$ are as follows: per RAM, $\mathcal{S}_{t,1}$ and $\mathcal{S}_{t,2}$ are such that $\mathcal{A}_t = \mathcal{S}_{t,1} \cup \mathcal{S}_{t,2}$. Additionally, due to their construction, $\mathcal{S}_{t,1} \cap \mathcal{S}_{t,2} = \emptyset$. Next, $\mathcal{B}^\star_{t,1}$ and $\mathcal{B}^\star_{t,2}$ are such that $\mathcal{B}^\star_{t,1} = \mathcal{B}^\star_{1:T} \cap \mathcal{S}_{t,1}$, and $\mathcal{B}^\star_2 = \mathcal{B}^\star_{1:T} \cap \mathcal{S}_{t,2}$; therefore, $\mathcal{B}^\star_{t,1} \cap \mathcal{B}^\star_{t,2} = \emptyset$ and $\mathcal{B}^\star_{1:T} = (\mathcal{B}^\star_{1,1} \cup \mathcal{B}^\star_{1,2}) \cup \cdots \cup (\mathcal{B}^\star_{T,1} \cup \mathcal{B}^\star_{T,2})$.

Superiority against worst-case failures: We focus on first row of Fig. 2, where β takes the values 3, 5, 7, and 9 (from left to right). For $\beta = 3$ (also, for $\beta = 1, 2$, accounting for the remaining, nondepicted simulations), RAM fares similar to `greedy`. In contrast, for the remaining values of β, RAM dominates `greedy`, achieving significantly lower error (observe the different scales among the subfigures for $\beta = 5, 7, 9$).

Across all β values in the Monte Carlo analysis (including those in Fig. 2), the suboptimality bound in (8) ranges from 0.02 to 0.10**, informing that RAM performs at least 2% to 10% the optimal. Specifically, c_f ranges from 0.89 to 0.98, whereas $f(\mathcal{A}_{1:t} \setminus \mathcal{B}^\star_{1:t})/f(\mathcal{M}_{1:t})$ remains again close to 0.95. Hence, the possible conservativeness of the bound stems from the conservativeness of its term $1 - c_f$.

Robustness against non-worst-case failures: We compare subfigure of Fig. 2 column-wise. Similarly to the autonomous navigation scenarios, the RAM's performance remains the same, or improves, against non-worst-case failures, and the lowest error is being observed against random failures. For example, if we focus on the rightmost column (where $\alpha = 10; \beta = 9$), at $t = 5$, then: for worst-case failures, RAM achieves error 611; in contrast, for greedy failures, RAM achieves the lower error 526; and for random failures, RAM achieves the even lower error 456. Generally, against greedy failures, RAM is again still superior to `greedy`; while against random failures, both have similar performance.

In summary, RAM remains superior even against system-wide failures, and even if the failures are non-worst-case.

V. Conclusion

We made the first step to adaptively protect critical control, estimation, and machine learning applications against sequential failures. Particularly, we focused on scenarios requiring the robust discrete optimization of systems per RSM. We provided RAM, the first online algorithm, which adapts to the history of failures, and guarantees a near-optimal performance even

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

against system-wide failures despite its minimal running time. To quantify the RAM's performance, we provided per-instance *a priori* bounds and tight, optimal *a posteriori* bounds. To this end, we used curvature notions, and contributed a first step toward characterizing the curvature's effect on the per-instance approximability of RSM. Our curvature-dependent bounds complement the current knowledge on the curvature's effect on the approximability of the failure-free optimization paradigm in (1) [23], [27], [30], [64]. Finally, we supported our theoretical results with numerical evaluations.

This article opens several avenues for future research. One is the decentralized implementation of RAM toward robust multi-agent autonomy and large-scale network design. And another is the extension of our results to optimization frameworks with general constraints (instead of cardinality, as in RSM), such as observability/controllability requirements, including matroid constraints, toward multirobot planning.

APPENDIX

In this Appendix, we provide all proofs. We use the notation

$$f(\mathcal{X} \mid \mathcal{X}') \triangleq f(\mathcal{X} \cup \mathcal{X}') - f(\mathcal{X}') \tag{18}$$

for any $\mathcal{X}, \mathcal{X}'$. Also, $\mathcal{X}_{1:t} \triangleq (\mathcal{X}_1, \ldots, \mathcal{X}_t)$ for any $\mathcal{X}_1, \ldots, \mathcal{X}_t$ (and $(\mathcal{X}_1, \ldots, \mathcal{X}_t) \equiv \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_t$).

A Preliminary lemmas

We list lemmas that support the proofs.

Lemma 18: Consider a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $f(\emptyset) = 0$. Then, for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$

$$f(\mathcal{A} \cup \mathcal{B}) \geq (1 - c_f) \left[f(\mathcal{A}) + f(\mathcal{B}) \right].$$

Proof of Lemma 18: Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$. Then,

$$f(\mathcal{A} \cup \mathcal{B}) = f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i \mid \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}). \tag{19}$$

The definition of c_f implies

$$\begin{aligned} & f(b_i \mid \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}) \\ \geq \quad & (1 - c_f)\, f(b_i \mid \{b_1, b_2, \ldots, b_{i-1}\}). \end{aligned} \tag{20}$$

The proof is completed by substituting (20) in (19), along with $f(\mathcal{A}) \geq (1 - c_f) f(\mathcal{A})$, since $c_f \leq 1$.

■

Lemma 19: Consider a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $f(\emptyset) = 0$. Then, for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$

$$f(\mathcal{A} \cup \mathcal{B}) \geq (1 - c_f) \left[f(\mathcal{A}) + \sum_{b \in \mathcal{B}} f(b) \right].$$

Proof of Lemma 19: Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$. Then

$$f(\mathcal{A} \cup \mathcal{B}) = f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i \mid \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}). \tag{21}$$

Now, c_fs Definition 7 implies

$$\begin{aligned} f(b_i \mid \mathcal{A} \cup \{b_1, b_2, \ldots, b_{i-1}\}) &\geq (1 - c_f) f(b_i \mid \emptyset) \\ &= (1 - c_f) f(b_i) \end{aligned} \tag{22}$$

where the latter holds since $f(\emptyset) = 0$. The proof is completed by substituting (22) in (21), along with $f(\mathcal{A}) \geq (1 - c_f) f(\mathcal{A})$, since $c_f \leq 1$.

■

Lemma 20: Consider a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $f(\emptyset) = 0$. Then, for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \setminus \mathcal{B} \neq \emptyset$

$$f(\mathcal{A}) + (1 - c_f) f(\mathcal{B}) \geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}) + f(\mathcal{A} \cap \mathcal{B}).$$

Proof of Lemma 20: Let $\mathcal{A} \setminus \mathcal{B} = \{i_1, i_2, \ldots, i_r\}$, where $r = |\mathcal{A} - \mathcal{B}|$. c_f's Definition 7 implies $f(i_j \mid (\mathcal{A} \cap \mathcal{B}) \cup \{i_1, i_2, \ldots, i_{j-1}\}) \geq (1 - c_f) f(i_j \mid \mathcal{B} \cup \{i_1, i_2, \ldots, i_{j-1}\})$, for any $i = 1, \ldots, r$. Summing the r inequalities

$$f(\mathcal{A}) - f(\mathcal{A} \cap \mathcal{B}) \geq (1 - c_f) \left[f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{B}) \right]$$

which implies the lemma.

■

Corollary 21: Consider a nondecreasing $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ such that $f(\emptyset) = 0$. Then, for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$

$$f(\mathcal{A}) + \sum_{b \in \mathcal{B}} f(b) \geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}).$$

Proof of Corollary 21: Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$. If $\mathcal{A} \neq \emptyset$, then

$$\begin{aligned} f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i) &\geq (1 - c_f) f(\mathcal{A}) + \sum_{i=1}^{|\mathcal{B}|} f(b_i) \\ &\geq (1 - c_f) f(\mathcal{A} \cup \{b_1\}) + \sum_{i=2}^{|\mathcal{B}|} f(b_i) \\ &\geq (1 - c_f) f(\mathcal{A} \cup \{b_1, b_2\}) + \sum_{i=3}^{|\mathcal{B}|} f(b_i) \\ &\vdots \\ &\geq (1 - c_f) f(\mathcal{A} \cup \mathcal{B}) \end{aligned} \tag{23}$$

where (23) holds since $0 \leq c_f \leq 1$, and the remaining inequalities are implied by applying Lemma 20 multiple times ($\mathcal{A} \cap \mathcal{B} = \emptyset$ implies $\mathcal{A} \setminus \{b_1\} \neq \emptyset$, $\mathcal{A} \cup \{b_1\} \setminus \{b_2\} \neq \emptyset$, …, $\mathcal{A} \cup \{b_1, b_2, \ldots, b_{|\mathcal{B}|-1}\} \setminus \{b_{|\mathcal{B}|}\} \neq \emptyset$).

If $\mathcal{A} = \emptyset$, then the proof follows the same reasoning as mentioned above but now we need to start from the following inequality, instead of (23)

$$\sum_{i=1}^{|\mathcal{B}|} f(b_i) \geq (1 - c_f) f(\{b_1\}) + \sum_{i=2}^{|\mathcal{B}|} f(b_i).$$

■

Lemma 22: Consider the sets $\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}$ selected by RAM's lines 3–4. Also, for all $t = 1, \ldots, T$, let $\mathcal{O}_t$ be any subset of $\mathcal{V}_t \setminus \mathcal{S}_{t,1}$ such that $|\mathcal{O}_t| \leq \alpha_t - \beta_t$. Then

$$f(\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{T,2}) \geq (1 - c_f)^2 f(\mathcal{O}_{1:T}). \tag{24}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

Proof of Lemma 22: For all $t = 1, \ldots, T$, let $\mathcal{R}_t \triangleq \mathcal{A}_t \setminus \mathcal{B}_t$; namely, $\mathcal{R}_t$ is the set that remains after the optimal (worst-case) removal $\mathcal{B}_t$ from $\mathcal{A}_t$. Furthermore, let $s^i_{t,2} \in \mathcal{S}_{t,2}$ denote the ith element added to $\mathcal{S}_{t,2}$ per RAM's lines 5–8, i.e., $\mathcal{S}_{t,2} = \{s^1_{t,2}, \ldots, s^{\alpha_t-\beta_t}_{t,2}\}$. Additionally, for all $i = 1, \ldots, \alpha_t - \beta_t$, denote $\mathcal{S}^i_{t,2} \triangleq \{s^1_{t,2}, \ldots, s^i_{t,2}\}$, and set $\mathcal{S}^0_{t,2} \triangleq \emptyset$. Next, order the elements in each $\mathcal{O}_t$ so that $\mathcal{O}_t = \{o^1_t, \ldots, o^{\alpha_t-\beta_t}_t\}$ and if $o^i_t \in \mathcal{S}_{t,2}$, then $o^i_t = s^i_{t,2}$, i.e., order the elements so that the common elements in $\mathcal{O}_t$ and $\mathcal{S}_{t,2}$ appear at the same index. Moreover, for all $i = 1, \ldots, \alpha_t - \beta_t$, denote $\mathcal{O}^i_t \triangleq \{o^1_t, \ldots, o^i_t\}$, and also set $\mathcal{O}^0_t \triangleq \emptyset$. Finally, let: $\mathcal{O}_{1:t} \triangleq (\mathcal{O}_1, \ldots, \mathcal{O}_t)$; $\mathcal{O}_{1:0} \triangleq \emptyset$; $\mathcal{S}_{1:t,2} \triangleq (\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{t,2})$; and $\mathcal{S}_{1:0,2} \triangleq \emptyset$. Then

$$f(\mathcal{O}_{1:T}) = \sum_{t=1}^{T} \sum_{i=1}^{\alpha_t-\beta_t} f(o^i_t \mid \mathcal{O}_{1:t-1} \cup \mathcal{O}^{i-1}_t) \tag{25}$$

$$\leq \frac{1}{1-c_f} \sum_{t=1}^{T} \sum_{i=1}^{\alpha_t-\beta_t} f(o^i_t \mid \mathcal{R}_{1:t-1} \cup \mathcal{S}^{i-1}_{t,2}) \tag{26}$$

$$\leq \frac{1}{1-c_f} \sum_{t=1}^{T} \sum_{i=1}^{\alpha_t-\beta_t} f(s^i_{t,2} \mid \mathcal{R}_{1:t-1} \cup \mathcal{S}^{i-1}_{t,2}) \tag{27}$$

$$\leq \frac{1}{(1-c_f)^2} \sum_{t=1}^{T} \sum_{i=1}^{\alpha_t-\beta_t} f(s^i_{t,2} \mid \mathcal{S}_{1:t-1,2} \cup \mathcal{S}^{i-1}_{t,2}) \tag{28}$$

$$= \frac{1}{(1-c_f)^2} f(\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{T,2}) \tag{29}$$

where (25) holds due to (18); (26) due to (4); (27) holds since $s^i_{t,2}$ is chosen greedily by the algorithm, given $\mathcal{R}_{1:t-1} \cup \mathcal{S}^{i-1}_{t,2}$; (28) holds for the same reasons as (26); (29) holds for the same reasons as (25).

■

Lemma 23: Consider the sets $\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}$ selected by RAM's lines 3–4. Also, for all $t = 1, \ldots, T$, let in Algorithm 2 be $\mathcal{K}_t = \mathcal{V}_t \setminus \mathcal{S}_{t,1}$ and $\delta_t = \alpha_t - \beta_t$. Finally, consider $\mathcal{P}_t$ such that $\mathcal{P}_t \subseteq \mathcal{K}_t$, $|\mathcal{P}_t| \leq \delta_t$, and $f(\mathcal{P}_{1:T})$ is maximal, that is,

$$\mathcal{P}_{1:T} \in \arg \max_{\bar{\mathcal{P}}_1 \subseteq \mathcal{K}_1, |\bar{\mathcal{P}}_1| \leq \delta_1} \cdots \max_{\bar{\mathcal{P}}_T \subseteq \mathcal{K}_T, |\bar{\mathcal{P}}_T| \leq \delta_T} f(\bar{\mathcal{P}}_{1:T}). \tag{30}$$

Then, $f(\mathcal{M}_{1:T}) \geq (1 - c_f) f(\mathcal{P}_{1:T})$.

Proof of Lemma 23: The proof is the same as that of [23, Th. 6].

■

Corollary 24: Consider the sets $\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}$ selected by RAM's lines 3–4, as well as, the sets $\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{T,2}$ selected by RAM's lines 5–8. Finally, consider $\mathcal{K}_t = \mathcal{V}_t \setminus \mathcal{S}_{t,1}$ and $\delta_t = \alpha_t - \beta_t$, and $\mathcal{P}_t$ per (30). Then,

$$f(\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{T,2}) \geq (1 - c_f)^3 f(\mathcal{P}_{1:T}).$$

Proof of Corollary 24: Let $\mathcal{O}_t = \mathcal{M}_t$ in (24). Then,

$$f(\mathcal{S}_{1,2}, \ldots, \mathcal{S}_{T,2}) \geq (1 - c_f)^2 f(\mathcal{M}_{1:T}). \tag{31}$$

Using in (31) Lemma 23, the proof is complete.

■

Lemma 25: Per the notation in Corollary 24, for all $t = 1, \ldots, T$, consider $\mathcal{K}_t = \mathcal{V}_t \setminus \mathcal{S}_{t,1}$, $\delta_t = \alpha_t - \beta_t$, and $\mathcal{P}_t$ per (30). Then, it holds true that

$$f(\mathcal{P}_{1:T}) \geq f^\star. \tag{32}$$

Proof of Lemma 25: We use the notation

$$h(\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}) \triangleq \max_{\bar{\mathcal{P}}_1 \subseteq \mathcal{V}_1 \setminus \mathcal{S}_{1,1}, |\bar{\mathcal{P}}_1| \leq \delta_1} \cdots \max_{\bar{\mathcal{P}}_T \subseteq \mathcal{V}_T \setminus \mathcal{S}_{T,1}, |\bar{\mathcal{P}}_T| \leq \delta_T} f(\bar{\mathcal{P}}_{1:T}). \tag{33}$$

For any $\hat{\mathcal{P}}_1, \ldots, \hat{\mathcal{P}}_T$ such that $\hat{\mathcal{P}}_t \subseteq \mathcal{V}_t \setminus \mathcal{S}_{t,1}$ and $|\hat{\mathcal{P}}_t| \leq \delta_t$ (for all $t = 1, \ldots, T$), hs definition in (33) implies

$$h(\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}) \geq f(\hat{\mathcal{P}}_1, \ldots, \hat{\mathcal{P}}_T). \tag{34}$$

Since (34) holds for any $\hat{\mathcal{P}}_T$ such that $\hat{\mathcal{P}}_T \subseteq \mathcal{V}_T \setminus \mathcal{S}_{T,1}$ and $|\hat{\mathcal{P}}_T| \leq \delta_T$, then it also holds for the $\hat{\mathcal{P}}_T$ that maximizes the right-hand side of (34), i.e.,

$$h(\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T,1}) \geq \max_{\bar{\mathcal{P}}_T \subseteq \mathcal{V}_T \setminus \mathcal{S}_{T,1}, |\bar{\mathcal{P}}_T| \leq \delta_T} f(\hat{\mathcal{P}}_{1:T-1}, \bar{\mathcal{P}}_T). \tag{35}$$

Treating in (35), the set $\mathcal{S}_{T,1}$ as a free variable, since (35) holds for any $\mathcal{S}_{T,1} \subseteq \mathcal{V}_T$ such that $|\mathcal{S}_{T,1}| \leq \beta_T$, then

$$\min_{\bar{\mathcal{B}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{B}}_T| \leq \beta_T} h(\mathcal{S}_{1,1}, \ldots, \mathcal{S}_{T-1,1}, \bar{\mathcal{B}}_T) \geq \min_{\bar{\mathcal{B}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{B}}_T| \leq \beta_T} \max_{\bar{\mathcal{P}}_T \subseteq \mathcal{V}_T \setminus \bar{\mathcal{B}}_T, |\bar{\mathcal{P}}_T| \leq \delta_T} f(\hat{\mathcal{P}}_{1:T-1}, \bar{\mathcal{P}}_T). \tag{36}$$

Specifically, (36) holds true for the same reason the following holds true: given a set function $f_1 : \mathcal{I} \mapsto \mathbb{R}$ [representing the left-hand side of (35)] and a set function $f_2 : \mathcal{I} \mapsto \mathbb{R}$ [representing the right-hand side of (35)], where $\mathcal{I} = \{\mathcal{S} : \mathcal{S} \subseteq \mathcal{V}_T, |\mathcal{S}| \leq \beta_T\}$, if $f_1(\mathcal{S}) \geq f_2(\mathcal{S})$ for every $\mathcal{S} \in \mathcal{I}$ [as (35) defines], then also the minimum of f_1 must be greater or equal to the minimum of f_2, i.e., $\min_{\mathcal{S} \in \mathcal{I}} f_1(\mathcal{S}) \geq \min_{\mathcal{S} \in \mathcal{I}} f_2(\mathcal{S})$. The reason if $\mathcal{S}^\star_1 \in \arg\min_{\mathcal{S} \in \mathcal{I}} f_1(\mathcal{S})$, then $f_1(\mathcal{S}^\star_1) \geq f_2(\mathcal{S}^\star_1)$, since $f_1(\mathcal{S}) \geq f_2(\mathcal{S})$ for every $\mathcal{S} \in \mathcal{I}$; but also $f_2(\mathcal{S}^\star_1) \geq \min_{\mathcal{S} \in \mathcal{I}} f_2(\mathcal{S})$, and, as a result, indeed, $\min_{\mathcal{S} \in \mathcal{I}} f_1(\mathcal{S}) \geq \min_{\mathcal{S} \in \mathcal{I}} f_2(\mathcal{S})$. Changing the symbol of the dummy variable $\mathcal{S}$ to $\bar{\mathcal{B}}_T$, we get (36).

Denote now the right-hand side of (36) by $z(\hat{\mathcal{P}}_{1:T-1})$. Since $\delta_T = \alpha_T - \beta_T$, and for $\bar{\mathcal{P}}_T$ in (36), it is $\bar{\mathcal{P}}_T \subseteq \mathcal{V}_T \setminus \bar{\mathcal{B}}_T$ and $|\bar{\mathcal{P}}_T| \leq \delta_T$, then it equivalently holds

$$z(\hat{\mathcal{P}}_{1:T-1}) = \min_{\bar{\mathcal{B}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{B}}_T| \leq \beta_T} \max_{\bar{\mathcal{A}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{A}}_T| \leq \alpha_T} f(\hat{\mathcal{P}}_{1:T-1}, \bar{\mathcal{A}}_T \setminus \bar{\mathcal{B}}_T). \tag{37}$$

Let in (37) $w(\bar{\mathcal{A}}_T \setminus \bar{\mathcal{B}}_T) \triangleq f(\hat{\mathcal{P}}_{1:T-1}, \bar{\mathcal{A}}_T \setminus \bar{\mathcal{B}}_T)$; we prove that the following holds true:

$$z(\hat{\mathcal{P}}_{1:T-1}) \geq \max_{\bar{\mathcal{A}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{A}}_T| \leq \alpha_T} \min_{\bar{\mathcal{B}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{B}}_T| \leq \beta_T} w(\bar{\mathcal{A}}_T \setminus \bar{\mathcal{B}}_T). \tag{38}$$

Particularly, for any $\hat{\mathcal{A}}_T \subseteq \mathcal{V}_T$ such that $|\hat{\mathcal{A}}_T| \leq \alpha_T$, and any $\hat{\mathcal{S}}_{T,1} \subseteq \mathcal{V}_T$ such that $|\hat{\mathcal{S}}_{T,1}| \leq \beta_T$, it is

$$\max_{\bar{\mathcal{A}}_T \subseteq \mathcal{V}_T, |\bar{\mathcal{A}}_T| \leq \alpha_T} w(\bar{\mathcal{A}}_T \setminus \hat{\mathcal{S}}_{T,1}) \geq w(\hat{\mathcal{A}}_T \setminus \hat{\mathcal{S}}_{T,1}). \tag{39}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

From (39), following the same reasoning as for the derivation of (36) from (35), considering in (39), the $\hat{\mathcal{S}}_{T,1}$ to be the free variable, we get

$$\min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T}\ \max_{\bar{\mathcal{A}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{A}}_T|\leq\alpha_T} w(\bar{\mathcal{A}}_T\setminus\bar{\mathcal{B}}_T) \geq \min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} w(\hat{\mathcal{A}}_T\setminus\bar{\mathcal{B}}_T). \quad (40)$$

Now, (40) implies (38). The reason (40) holds for any $\bar{\mathcal{A}}_T\subseteq\mathcal{V}_T$ such that $|\bar{\mathcal{A}}_T|\leq\alpha_T$, while the left-hand side of (40) is equal to $z(\hat{\mathcal{P}}_{1:T-1})$, which is independent of $\hat{\mathcal{A}}_T$; therefore, if we maximize the right-hand side of (40) with respect to $\bar{\mathcal{A}}_T$, then indeed we get (38).

All in all, due to (38), (36) becomes

$$\min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} h(\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-1,1},\bar{\mathcal{B}}_T) \geq \max_{\bar{\mathcal{A}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{A}}_T|\leq\alpha_T}\ \min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} f(\hat{\mathcal{P}}_{1:T-1},\bar{\mathcal{A}}_T\setminus\bar{\mathcal{B}}_T). \quad (41)$$

The left-hand side of (41) is a function of $\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-1,1}$; denote it as $h'(\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-1,1})$. Similarly, the right-hand side of (41) is a function of $\hat{\mathcal{P}}_{1:T-1}$; denote it as $f'(\hat{\mathcal{P}}_{1:T-1})$. Given these notations, (41) is equivalently written as

$$h'(\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-1,1}) \geq f'(\hat{\mathcal{P}}_{1:T-1}) \quad (42)$$

which has the same form as (34). Therefore, by following the same steps as those we used from (34) and onward, we similarly get

$$\min_{\bar{\mathcal{B}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{B}}_{T-1}|\leq\beta_{T-1}} h'(\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-2,1},\bar{\mathcal{B}}_{T-1}) \geq \max_{\bar{\mathcal{A}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{A}}_{T-1}|\leq\alpha_{T-1}}\ \min_{\bar{\mathcal{B}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{B}}_{T-1}|\leq\beta_{T-1}} f'(\hat{\mathcal{P}}_{1:T-2},\bar{\mathcal{A}}_{T-1}\setminus\bar{\mathcal{B}}_{T-1}) \quad (43)$$

which, given the definitions of $h'(\cdot)$ and $f'(\cdot)$, is equivalent to

$$\min_{\bar{\mathcal{B}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{B}}_{T-1}|\leq\beta_{T-1}}\ \min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} h(\mathcal{S}_{1,1},\ldots,\mathcal{S}_{T-2,1},\bar{\mathcal{B}}_{T-1},\bar{\mathcal{B}}_T) \geq \max_{\bar{\mathcal{A}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{A}}_{T-1}|\leq\alpha_{T-1}}\ \min_{\bar{\mathcal{B}}_{T-1}\subseteq\mathcal{V}_{T-1},|\bar{\mathcal{B}}_{T-1}|\leq\beta_{T-1}} \max_{\bar{\mathcal{A}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{A}}_T|\leq\alpha_T}\ \min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} f(\hat{\mathcal{P}}_{1:T-2},\bar{\mathcal{A}}_{T-1}\setminus\bar{\mathcal{B}}_{T-1},\bar{\mathcal{A}}_T\setminus\bar{\mathcal{B}}_T).$$

Equation (43) has the same form as (41). Therefore, repeating the same steps as mentioned above for another $T-2$ times [starting now from (43) instead of (41)], we get

$$\min_{\bar{\mathcal{B}}_1\subseteq\mathcal{V}_1,|\bar{\mathcal{B}}_1|\leq\beta_1}\cdots\min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} h(\bar{\mathcal{B}}_{1:T}) \geq \max_{\bar{\mathcal{A}}_1\subseteq\mathcal{V}_1,|\bar{\mathcal{A}}_1|\leq\alpha_1}\ \min_{\bar{\mathcal{B}}_1\subseteq\mathcal{V}_1,|\bar{\mathcal{B}}_1|\leq\beta_1}\cdots\max_{\bar{\mathcal{A}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{A}}_T|\leq\alpha_T}\ \min_{\bar{\mathcal{B}}_1\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} f(\bar{\mathcal{A}}_1\setminus\bar{\mathcal{B}}_1,\ldots,\bar{\mathcal{A}}_T\setminus\bar{\mathcal{B}}_T) \quad (44)$$

which implies (32) since the right-hand side of (44) is equal to the right-hand side of (32), while for the left-hand side of (44), the following holds:

$$\min_{\bar{\mathcal{B}}_1\subseteq\mathcal{V}_1,|\bar{\mathcal{B}}_1|\leq\beta_1}\cdots\min_{\bar{\mathcal{B}}_T\subseteq\mathcal{V}_T,|\bar{\mathcal{B}}_T|\leq\beta_T} h(\bar{\mathcal{B}}_{1:T}) \leq f(\mathcal{P}_{1:T}).$$

■

B Proof of Proposition 2

We compute the running time of RAM's line 3 and lines 5–8. Line 3 needs $|\mathcal{V}_t|\tau_f+|\mathcal{V}_t|\log(|\mathcal{V}_t|)+|\mathcal{V}_t|+O(\log(|\mathcal{V}_t|))$ time: it asks for $|\mathcal{V}_t|$ evaluations of f, and their sorting, which takes $|\mathcal{V}_t|\log(|\mathcal{V}_t|)+|\mathcal{V}_t|+O(\log(|\mathcal{V}_t|))$ time (using, e.g., the merge sort algorithm). Lines 5–8 need $(\alpha_t-\beta_t)[|\mathcal{V}_t|\tau_f+|\mathcal{V}_t|]$ time: the while loop is repeated $\alpha_t-\beta_t$ times, and during each loop at most $|\mathcal{V}_t|$ evaluations of f are needed (line 5), plus at most $|\mathcal{V}_t|$ steps for a maximal element to be found (line 6). Hence, RAM runs at each t in $(\alpha_t-\beta_t)[|\mathcal{V}_t|\tau_f+|\mathcal{V}_t|]+|\mathcal{V}_t|\log(|\mathcal{V}_t|)+|\mathcal{V}_t|+O(\log(|\mathcal{V}_t|))=O(|\mathcal{V}_t|(\alpha_t-\beta_t)\tau_f)$ time.

C Proof of Theorem 10

We first prove (6) and then (5). We use the notation as follows:

1) $\mathcal{S}^+_{t,1}\triangleq\mathcal{S}_{t,1}\setminus\mathcal{B}^\star_t$, i.e., $\mathcal{S}^+_{t,1}$ is the remaining set after the optimal (worst-case) removal $\mathcal{B}^\star_t$;
2) $\mathcal{S}^+_{t,2}\triangleq\mathcal{S}_{t,2}\setminus\mathcal{B}^\star_t$;
3) $\mathcal{P}_{1:T}$ be a solution to (30).

Proof of (6): For $T>1$, we have

$$\begin{aligned} f(\mathcal{A}_{1:T}\setminus\mathcal{B}^\star_{1:T}) &= f(\mathcal{S}^+_{1,1}\cup\mathcal{S}^+_{1,2},\ldots,\mathcal{S}^+_{T,1}\cup\mathcal{S}^+_{T,2}) && (45)\\ &\geq (1-c_f)\sum_{t=1}^{T}\sum_{v\in\mathcal{S}^+_{t,1}\cup\mathcal{S}^+_{t,2}} f(v) && (46)\\ &\geq (1-c_f)\sum_{t=1}^{T}\sum_{v\in\mathcal{S}_{t,2}} f(v) && (47)\\ &\geq (1-c_f)^2 f(\mathcal{S}_{1,2},\ldots,\mathcal{S}_{T,2}) && (48)\\ &\geq (1-c_f)^5 f(\mathcal{P}_{1:T}) && (49)\\ &\geq (1-c_f)^5 f^\star && (50) \end{aligned}$$

where (45) follows from the definitions of $\mathcal{S}^+_{t,1}$ and $\mathcal{S}^+_{t,2}$; (46) follows from (45), due to Lemma 19; (47) follows from (46), because: for all $v\in\mathcal{S}^+_{t,1}$ and $v'\in\mathcal{S}_{t,2}\setminus\mathcal{S}^+_{t,2}$, it is $f(v)\geq f(v')$, and $\mathcal{S}_{t,2}=(\mathcal{S}_{t,2}\setminus\mathcal{S}^+_{t,2})\cup\mathcal{S}^+_{t,2}$; (48) follows from (47) due to Corollary 21; (49) follows from (48) due to Corollary 24; finally, (50) follows from (49) due to Lemma 25.

For $T=1$, the proof follows the same steps up to (48), at which point $f(\mathcal{S}_{1,2})\geq(1-c_f)f(\mathcal{P}_1)$ instead, due to Lemma 23 (since $\mathcal{S}_{1,2}=\mathcal{M}_1$). ■

Proof of (5): For $T>1$, we follow similar steps

$$f(\mathcal{A}_{1:T}\setminus\mathcal{B}^\star_{1:T}) = f(\mathcal{S}^+_{1,1}\cup\mathcal{S}^+_{1,2},\ldots,\mathcal{S}^+_{T,1}\cup\mathcal{S}^+_{T,2}) \quad (51)$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

$$\geq (1-\kappa_f)\sum_{t=1}^{T}\sum_{v\in\mathcal{S}_{t,1}^{+}\cup\mathcal{S}_{t,2}^{+}} f(v) \tag{52}$$

$$\geq (1-\kappa_f)\sum_{t=1}^{T}\sum_{v\in\mathcal{S}_{t,2}} f(v) \tag{53}$$

$$\geq (1-\kappa_f) f(\mathcal{S}_{1,2},\ldots,\mathcal{S}_{T,2}) \tag{54}$$

$$\geq (1-\kappa_f)^4 f(\mathcal{P}_{1:T}) \tag{55}$$

$$\geq (1-\kappa_f)^4 f^\star \tag{56}$$

where (51) follows from the definitions of $\mathcal{S}_{t,1}^{+}$ and $\mathcal{S}_{t,2}^{+}$; (52) follows from (51) due to Lemma 18 and the fact that $c_f = \kappa_f$ for f being submodular; (53) follows from (52) because for all $v \in \mathcal{S}_{t,1}^{+}$ and $v' \in \mathcal{S}_{t,2} \setminus \mathcal{S}_{t,2}^{+}$, it is $f(v) \geq f(v')$, while $\mathcal{S}_{t,2} = (\mathcal{S}_{t,2} \setminus \mathcal{S}_{t,2}^{+}) \cup \mathcal{S}_{t,2}^{+}$; (54) follows from (53) because f is submodular and, as a result, $f(\mathcal{S}) + f(\mathcal{S}') \geq f(\mathcal{S} \cup \mathcal{S}')$, for any $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{V}$ [63, Proposition 2.1]; (55) follows from (54) due to Corollary 24, along with the fact that since f is monotone submodular it is $c_f = \kappa_f$; finally, (56) follows from (55) due to Lemma 25.

For $T = 1$, the proof follows the same steps up to (54), at which point $f(\mathcal{S}_{1,2}) \geq 1/\kappa_f(1-e^{-\kappa_f}) f(\mathcal{P}_1)$, due to [27, Th. 5.4]. ■

D Proof of Theorem 13

To prove (7), we have

$$\begin{aligned} & f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^{\star}) \\ &= f(\mathcal{M}_{1:t}) \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^{\star})}{f(\mathcal{M}_{1:t})} \\ &\geq \begin{cases} \frac{1-e^{-\kappa_f}}{\kappa_f} \frac{f(\mathcal{A}_1 \setminus \mathcal{B}_1^{\star})}{f(\mathcal{M}_1)} f(\mathcal{P}_1), & t = 1 \\ \frac{1}{1+\kappa_f} \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^{\star})}{f(\mathcal{M}_{1:t})} f(\mathcal{P}_{1:t}), & t > 1 \end{cases} \qquad (57) \\ &\geq \begin{cases} \frac{1-e^{-\kappa_f}}{\kappa_f} \frac{f(\mathcal{A}_1 \setminus \mathcal{B}_1^{\star})}{f(\mathcal{M}_1)} f_1^{\star}, & t = 1 \\ \frac{1}{1+\kappa_f} \frac{f(\mathcal{A}_{1:t} \setminus \mathcal{B}_{1:t}^{\star})}{f(\mathcal{M}_{1:t})} f_t^{\star}, & t > 1 \end{cases} \qquad (58) \end{aligned}$$

where (57) holds since [27, Th. 5.4] implies $f(\mathcal{M}_1) \geq 1/\kappa_f(1-e^{-\kappa_f}) f(\mathcal{P}_1)$, while [27, Th. 2.3] implies $f(\mathcal{M}_{1:t}) \geq 1/(1+\kappa_f) f(\mathcal{P}_{1:t})$. Finally, (58) is proved following the same steps as in Lemma 25 proof.

The proof of (8) follows similar steps as mentioned above but it is based instead on Lemma 23.

E Proof of Theorem 14

It can be verified that for $t = 1$ (7) is tight for any $\beta_t \leq \alpha_t$ for the families of functions in [27, Th. 5.4], and for $t > 1$, it is tight for the families of functions in [27, Th. 2.12]. Similarly, it can be verified (8) is optimal for the families of functions in [23, Th. 8] for $\alpha_t = |\mathcal{V}_t|^{1/2}$ and any $\beta_t \leq \alpha_t - |\mathcal{V}_t|^{1/3}$.

Acknowledgment

The authors would like to thank R. Vohra, University of Pennsylvania, and L. Carlone, Massachusetts Institute of Technology for inspiring comments and discussions.

References

[1] T. Abdelzaher *et al.*, "Will distributed computing revolutionize peace? The emergence of Battlefield IoT," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 1129–1138.
[2] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 1–20, Feb. 2019.
[3] T. He *et al.*, "Achieving real-time target tracking usingwireless sensor networks," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, 2006, pp. 37–48.
[4] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2019.
[5] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.
[6] 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.
[7] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, *Submodularity in Dynamics and Control of Networked Systems*. Berlin, Germany: Springer, 2016.
[8] V. Tzoumas, N. A. Atanasov, A. Jadbabaie, and G. J. Pappas, "Scheduling nonlinear sensors for stochastic process estimation," in *Proc. Amer. Control Conf.*, 2017, pp. 580–585.
[9] 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.
[10] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "LQG control and sensing co-design," in *Proc. IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2020.2997661.
[11] Y. Zhao, F. Pasqualetti, and J. Cortés, "Scheduling of control nodes for improved network controllability," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 1859–1864.
[12] E. Nozari, F. Pasqualetti, and J. Cortés, "Time-invariant versus time-varying actuator scheduling in complex networks," in *Proc. Amer. Control Conf.*, 2017, pp. 4995–5000.
[13] T. Ikeda and K. Kashima, "Sparsity-constrained controllability maximization with application to time-varying control node selection," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 321–326, Jul. 2018.
[14] J. Leskovec *et al.*, "Cost-effective outbreak detection in networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 420–429.
[15] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[16] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Control Netw. Syst.*, vol. 3, no. 1, pp. 91–101, Mar. 2016.
[17] U. Feige, "A threshold of $ln(n)$ for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.
[18] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," in *Polyhedral Combinatorics*. Berlin, Germany: Springer, 1978, pp. 73–87.
[19] D. Foster, H. Karloff, and J. Thaler, "Variable selection is hard," in *Proc. Conf. Learn. Theory*, 2015, pp. 696–709.
[20] L. Ye, S. Roy, and S. Sundaram, "On the complexity and approximability of optimal sensor selection for kalman filtering," in *Proc. Amer. Control Conf.*, 2018, pp. 5049–5054.
[21] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1057–1064.
[22] Z. Wang, B. Moran, X. Wang, and Q. Pan, "Approximation for maximizing monotone non-decreasing set functions with a greedy method," *J. Combinatorial Optim.*, vol. 31, no. 1, pp. 29–43, 2016.
[23] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Math. Oper. Res.*, vol. 42, no. 4, pp. 1197–1218, 2017.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

[24] T. Abdelzaher *et al.*, "Toward an internet of battlefield things: A resilience perspective," *Computer*, vol. 51, no. 11, pp. 24–36, 2018.

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

[26] R. B. Myerson, *Game Theory*. Cambridge, MA, USA: Harvard Univ. Press, 2013.

[27] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm," *Discr. Appl. Math.*, vol. 7, no. 3, pp. 251–274, 1984.

[28] R. K. Iyer, S. Jegelka, and J. A. Bilmes, "Curvature and optimal algorithms for learning and minimizing submodular functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2742–2750.

[29] A. Krause and V. Cevher, "Submodular dictionary selection for sparse representation," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 567–574.

[30] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," 2013, *arXiv:1311.4728*.

[31] J. B. Orlin, A. S. Schulz, and R. Udwani, "Robust monotone submodular function maximization," in *Proc. Int. Conf. Integer Program. Combinatorial Optim.*, 2016, pp. 312–324.

[32] I. Bogunovic, S. Mitrović, J. Scarlett, and V. Cevher, "Robust submodular maximization: A non-uniform partitioning approach," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 508–516.

[33] V. Tzoumas, K. Gatsis, A. Jadbabaie, and G. J. Pappas, "Resilient monotone submodular function maximization," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 1362–1367.

[34] A. Rahmattalabi, P. Vayanos, and M. Tambe, "A robust optimization approach to designing near-optimal strategies for constant-sum monitoring games," in *Proc. Int. Conf. Decis. Game Theory Secur.*, 2018, pp. 603–622.

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

[36] I. Bogunovic, J. Zhao, and V. Cevher, "Robust maximization of non-submodular objectives," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 890–899.

[37] B. Schlotfeldt, V. Tzoumas, D. Thakur, and G. J. Pappas, "Resilient active information gathering with mobile robots," *in Proc. IEEE/ RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4309–4316.

[38] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 129–136, Jan. 2019.

[39] S. Mitrovic, I. Bogunovic, A. Norouzi-Fard, J. M. Tarnawski, and V. Cevher, "Streaming robust submodular maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4560–4569.

[40] B. Mirzasoleiman, A. Karbasi, and A. Krause, "Deletion-robust submodular maximization: Data summarization with 'the right to be forgotten'," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2449–2458.

[41] E. Kazemi, M. Zadimoghaddam, and A. Karbasi, "Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 2544–2553.

[42] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, Berlin, Germany: Springer, 2006.

[43] L. F. Cómbita, A. A. Córdenas, and N. Quijano, "Mitigating sensor attacks against industrial control systems," *IEEE Access*, vol. 7, pp. 92 444–92 455, 2019.

[44] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, 2011, Art. no. 31.

[45] M. Jin, J. Lavaei, and K. H. Johansson, "Power grid ac-based state estimation: Vulnerability analysis against cyber attacks," *IEEE Trans. Autom. Control*, vol. 64, no. 5, pp. 1784–1799, May 2019.

[46] A. Clark and L. Niu, "Linear quadratic Gaussian control under false data injection attacks," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 5737–5743.

[47] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 46–65, Feb. 2015.

[48] X. Jin, W. M. Haddad, and T. Yucelen, "An adaptive control architecture for mitigating sensor and actuator attacks in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 6058–6064, Nov. 2017.

[49] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[50] L. S. Perelman, W. Abbas, X. Koutsoukos, and S. Amin, "Sensor placement for fault location identification in water networks: A minimum test cover approach," *Automatica*, vol. 72, pp. 166–176, 2016.

[51] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 4917–4932, Oct. 2017.

[52] M. Pajic *et al.*, "Robustness of attack-resilient state estimators," in *Proc. ACM/IEEE 5th Int. Conf. Cyber-Phys. Syst.*, 2014, pp. 163–174.

[53] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1029–1043, Mar. 2020.

[54] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 93–109, Feb. 2015.

[55] J. Usevitch and D. Panagou, "Resilient leader-follower consensus to arbitrary reference values in time-varying graphs," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1755–1762, Apr. 2020.

[56] G. De La Torre, T. Yucelen, and J. D. Peterson, "Resilient networked multiagent systems: A distributed adaptive control approachy," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 5367–5372.

[57] L. Su and S. Shahrampour, "Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements," in *Proc. IEEE Trans. Autom. Control*, Sep. 2020, vol. 65, no. 9, pp. 3758–3771, doi: 10.1109/TAC.2019.2951686.

[58] Y. Chen, S. Kar, and J. Moura, "Resilient distributed estimation: Sensor attacks," *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3772–3779, Sep. 2019.

[59] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks,"*Proc. IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.

[60] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Proc. Int. Workshop Hybrid Syst.: Comput. Control*, 2009, pp. 31–45.

[61] A.-Y. Lu and G.-H. Yang, "Input-to-state stabilizing control for cyber-physical systems with multiple transmission channels under denial of service," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1813–1820, Jun. 2018.

[62] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Resilient monotone sequential maximization," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 7261–7268.

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

[64] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games Econ. Behav.*, vol. 55, no. 2, pp. 270–296, 2006.

[65] E. R. Elenberg *et al.*, "Restricted strong convexity implies weak submodularity," *Ann. Statist.*, vol. 46, no. 6B, pp. 3539–3568, 2018.

[66] L. F. Chamon and A. Ribeiro, "Near-optimality of greedy set selection in the sampling of graph signals," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2016, pp. 1265–1269.

[67] B. Guo, O. Karaca, T. Summers, and M. Kamgarpour, "Actuator placement for optimizing network performance under controllability constraints," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 7140–7147.

[68] D. Sharma, A. Kapoor, and A. Deshpande, "On greedy maximization of entropy," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1330–1338.

[69] L. F. Chamon, G. J. Pappas, and A. Ribeiro, "The mean square error in kalman filtering sensor selection is approximately supermodular," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 343–350.

[70] R. Khanna, E. Elenberg, A. Dimakis, S. Negahban, and J. Ghosh, "Scalable greedy feature selection via weak submodularity," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1560–1568.

[71] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.

[72] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," *J. Combinatorial Theory B*, vol. 80, no. 2, pp. 346–355, 2000.

[73] S. Iwata and J. B. Orlin, "A simple combinatorial algorithm for submodular function minimization," in *Proc. ACM-SIAM Symp. Discr. Algorithms*, 2009, pp. 1230–1237.

[74] Y. T. Lee, A. Sidford, and S. C.-W. Wong, "A faster cutting plane method and its implications for combinatorial and convex optimization," in *Proc. IEEE 56th Annu. Symp. Found. Comput. Sci.*, 2015, pp. 1049–1065.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.

[75] D. Chakrabarty, Y. T. Lee, A. Sidford, and S. C.-W. Wong, "Subquadratic submodular function minimization," in *Proc. 49th Annu. ACM SIGACT Symp. Theory Comput.*, 2017, pp. 1220–1231.
[76] M. E. Halabi and S. Jegelka, "Optimal approximation for unconstrained non-submodular minimization," in *Proc. Int. Conf. Mach. Learn.*, Nov. 2020, pp. 3961–3972.
[77] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ Press, 2004.
[78] L. Ye, S. Roy, and S. Sundaram, "Resilient sensor placement for kalman filtering in networked systems: Complexity and algorithms," *IEEE Control Netw. Syst.*, vol. 7, no. 4, pp. 1870–1881, Dec. 2020.
[79] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *J. Mach. Learn. Res.*, vol. 9, pp. 2761–2801, 2008.
[80] A. Rahmattalabi *et al.*, "Exploring algorithmic fairness in robust graph covering problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15 776–15 787.
[81] D. Lu, Y. Qu, F. Wu, H. Dai, C. Dong, and G. Chen, "Robust server placement for edge computing," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2020, pp. 285–294.
[82] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Säkkä, "Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Auton. Robots*, vol. 39, no. 3, pp. 221–238, 2015.



Vasileios Tzoumas (Member, IEEE) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2012, the M.A. degree in statistics from the Wharton School of Business, University of Pennsylvania, Philadelphia, PA, USA, in 2016, the M.Sc. degree in electrical engineering from the University of Pennsylvania, in 2016, and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania in 2018

He is currently an Assistant Professor with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA. Previously, he was with the Laboratory for Information and Decision Systems (LIDS), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, were he was a Research Scientist (2019–2020), and a Postdoctoral Associate (2018–2019). He was a Visiting Ph.D. Student with the Institute for Data, Systems, and Society (IDSS), MIT during 2017. He works on control, learning, and perception, as well as combinatorial and distributed optimization, with applications to robotics, cyber-physical systems, and self-reconfigurable aerospace systems. He aims for trustworthy collaborative autonomy. His research work includes foundational results on robust and adaptive combinatorial optimization, with applications to multirobot information gathering for resiliency against robot failures and adversarial removals.

Dr. Tzoumas is a recipient of the Best Paper Award in Robot Vision at the 2020 IEEE International Conference on Robotics and Automation (ICRA), and was a Best Student Paper Award Finalist at the 2017 IEEE Conference in Decision and Control (CDC).



Ali Jadbabaie (Fellow, IEEE) received the B.S. degree (with high honors) from the Sharif University of Technology, Tehran, Iran, in 1995, the M.S. degree in electrical and computer engineering from the University of New Mexico, Albuquerque, NM, USA, in 1997, and the Ph.D. degree in control and dynamical systems from the California Institute of Technology, Pasadena, CA, USA, in 2000.

He is the JR East Professor of Engineering and an Associate Director of the Institute for Data, Systems, and Society, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where he is also on the faculty of the Department of Civil and Environmental Engineering and a Principal Investigator with the Laboratory for Information and Decision Systems (LIDS). He is the Director of the Sociotechnical Systems Research Center, one of MIT's 13 laboratories. He was a Postdoctoral Scholar with Yale University, New Haven, CT, USA, before joining the faculty at Penn in July 2002. Prior to joining MIT faculty, he was the Alfred Fitler Moore a Professor of Network Science and held secondary appointments in computer and information science and operations, information and decisions at Wharton School, Philadelphia USA. His current research interests include the interplay of dynamic systems and networks with specific emphasis on multiagent coordination and control, distributed optimization, network science, and network economics.

Dr. Jadbabaie was the inaugural Editor-in-Chief of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, a new interdisciplinary journal sponsored by several IEEE societies. He is a recipient of a National Science Foundation Career Award, an Office of Naval Research Young Investigator Award, the O. Hugo Schuck Best Paper Award from the American Automatic Control Council, and the George S. Axelby Best Paper Award from the IEEE Control Systems Society. His students have been winners and finalists of student best paper awards at various American Control Conference and Conference in Decision and Control conferences. He is the recipient of the Vannevar Bush Fellowship from the office of Secretary of Defense.



George J. Pappas (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998.

He is currently the Joseph Moore Professor and the Chair of the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, USA. He is a Member of the GRASP Lab and the PRECISE Center. He had previously served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, cyberphysical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks.

Dr. Pappas has been the recipient of various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation PECASE Award and numerous best student papers awards.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:55:44 UTC from IEEE Xplore. Restrictions apply.