

Randomized Greedy Methods for Weak Submodular Sensor Selection with Robustness Considerations

Ege C. Kaya, Michael Hibbard, Takashi Tanaka,
Ufuk Topcu, Abolfazl Hashemi*

Abstract

We study a pair of budget- and performance-constrained weak-submodular maximization problems. For computational efficiency, we explore the use of stochastic greedy algorithms which limit the search space via random sampling instead of the standard greedy procedure which explores the entire feasible search space. We propose a pair of stochastic greedy algorithms, namely, MODIFIED RANDOMIZED GREEDY (MRG) and DUAL RANDOMIZED GREEDY (DRG) to approximately solve the budget- and performance-constrained problems, respectively. For both algorithms, we derive approximation guarantees that hold with high probability. We then examine the use of DRG in robust optimization problems wherein the objective is to maximize the worst-case of a number of weak submodular objectives and propose the RANDOMIZED WEAK SUBMODULAR SATURATION ALGORITHM (RANDOM-WSSA). We further derive a high-probability guarantee for when RANDOM-WSSA successfully constructs a robust solution. Finally, we showcase the effectiveness of these algorithms in a variety of relevant uses within the context of Earth-observing low Earth orbit satellite constellations which estimate atmospheric weather conditions and provide Earth coverage.

Keywords: aerospace, decision making and autonomy, sensor-data fusion, distributed optimization for large-scale systems, large scale optimization problems, modeling and decision making in complex systems, probabilistic robustness, robustness analysis.

1 Introduction

In sensor selection problems, especially those involving an extensive network of sensors, one often needs to select an optimal subset for a given task rather than using all available due to restrictions on resources. Such restrictions, usually modeled in terms of constraints on the cardinality or the cost of feasible solutions, introduce the need to design principled decision-making algorithms for the creation of task-optimal selections of sensors that deliver high performance while adhering to the inherent resource, performance and robustness constraints. These problems occur naturally in many areas such as power system monitoring [3, 4], economics and algorithmic game theory [5–7], and finance [8]. In this work, we motivate the problem setting by considering low Earth orbit (LEO) constellations of Earth-observing satellites [9].

The problem of choosing the optimal set of sensors is a computationally challenging combinatorial optimization that is known to be NP-hard [10]. Because of this complexity and theoretical barrier, a considerable amount of research has delved into creating heuristic algorithms aimed at providing approximate solutions for the problem.

*Ege C. Kaya and Abolfazl Hashemi are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. Michael Hibbard, Takashi Tanaka and Ufuk Topcu are with the Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Austin, TX, USA. This work was presented in part at the 2023 American Control Conference (ACC) [1]. The full work was published in *Automatica*, 2025 [2]. This work is supported in part by DARPA grant HR00112220025.

Particularly well-known is the scenario where the objective can be formulated using a function that is both monotone nondecreasing and submodular, i.e., a set function that has the familiar property of demonstrating *diminishing marginal returns*. When sensor selection is made in the presence of a cardinality constraint on the number of sensors, a proven result is that the GREEDY algorithm achieves an approximation factor of $(1 - 1/e)$ in comparison to the optimal solution [11]. Specifically, at each iteration, the GREEDY algorithm considers each element in the set of remaining sensors, evaluates them in terms of marginal gain, and adds the element with the highest marginal gain to the set of selected elements.

Modified versions of GREEDY have been developed for addressing sensor selection problems with the more general case of budget constraints [12]. In this case, each sensor incurs a cost, and the decision-maker is restricted by a predetermined budget for their selections. The dual of this problem, where the objective is to identify the most cost-effective set of sensors to meet a specific performance constraint, has also been considered [13].

In situations involving a large number of sensors, the computational demands of GREEDY become impractical due to the necessity of evaluating the marginal return of every remaining sensor at each iteration. In [14], a computationally cheaper alternative is proposed, often referred to as the STOCHASTIC or RANDOMIZED GREEDY algorithm in the literature. This approach limits the function evaluations to a randomly sampled subset of the remaining sensors during each iteration and enjoys theoretical assurances concerning the expected performance of the constructed solution relative to the optimal solution. Expanding upon this randomized algorithm, [15] accounts for situations where the performance objective is defined by a weak submodular function, which is a function that is either truly submodular or whose maximum violation of submodularity is bounded. Moreover, the authors derive a theoretical guarantee that holds with high probability for the performance of the randomized greedy algorithm in the weak submodular regime. However, it is important to note that the setting they consider is restricted solely to the cardinality-constrained case and does not take into account the nontrivial generalization of this setting to one involving the presence of a budget constraint and nonuniform selection costs.

In this work, we draw inspiration from the previously aforementioned work to propose a pair of novel extensions to the RANDOMIZED GREEDY algorithm for use in budget- and performance-constrained problems, namely, the MODIFIED RANDOMIZED GREEDY (MRG) and DUAL RANDOMIZED GREEDY (DRG) algorithms, respectively. For both cases, we assume that the performance objective is characterized by a weak submodular function and that the sensor costs are additive and possibly nonuniform. To the best of our knowledge, the use of randomized greedy algorithms in these two scenarios is a novel approach. In the ensuing discussion, we further derive theoretical guarantees that hold with high probability on the approximation factors for both algorithms. As expected, in the case where the performance objectives are not weak submodular but truly submodular, and where the standard greedy procedure is used instead of a randomized greedy approach, both of the derived theoretical guarantees reduce to the standard results proven in the literature.

We further examine the use of the proposed DRG algorithm in the context of robust multi-objective submodular maximization where the robustness is captured by maximizing the value of the worst objective. This notion of worst-case robustness extends the setting explored in [16] which proposes the SUBMODULAR SATURATION ALGORITHM (SSA) in the presence of multiple monotone nondecreasing submodular objective functions. We propose a modified version of the algorithm, which we name RANDOMIZED WEAK SUBMODULAR SATURATION ALGORITHM (RANDOM-WSSA), and likewise derive a theoretical guarantee on the performance of the algorithm that holds with high probability.

The remainder of the paper is organized as follows: In Section 2, we motivate our work in the context of LEO satellite constellations while in Section 3 we provide the required preliminary

information used in the rest of the work. In Section 4, we introduce the MRG and the DRG algorithms and derive lower bounds on their approximation ratios. In Section 5, we introduce RANDOM-WSSA for the multi-objective robustness problem and provide theoretical guarantees on its success. Finally, in Section 6, we corroborate our theoretical results by providing numerical examples demonstrating the efficacy of the three proposed algorithms for sensor selection in Earth-observation tasks, as well as in scenarios involving multi-objective robustness.

2 Motivation: Satellite Sensing Networks

We start by presenting an illustrative motivating example within the field of LEO satellite constellations. Over the past few years, there has been a growing interest in satellite mission designs to use a cluster of simpler satellites rather than relying on a single intricate satellite. This trend has made the use of CubeSat-based designs [17] commonplace, such as the constellation depicted in Figure 1. Constellations built on such concepts provide mission designers with various advantages. These advantages include cost savings in the design and launch expenses [18], increased redundancy to address potential failures, and improved temporal resolution in observations due to more frequent revisits [19]. Some examples of missions that have achieved success with this design philosophy are the NASA TROPICS mission [20] and the Planet Labs Flock Constellation [21]. The NASA TROPICS mission had as its primary objective to acquire temporally dense observations of weather phenomena, while the objective of the Flock Constellation was high-resolution imaging of the Earth’s surface. Proposed uses for CubeSat-based missions include many types of Earth science-related applications [17,22], as well as disaster management, natural hazard response [23,24] and air traffic monitoring [25,26].

Naturally, as the number of LEO constellations, as well as the number of satellites in the constellation increases, more human operators are needed for their supervision and operation. One frequent mode of operation may involve the selection of only a subset of the satellites to use for sensing at each time step, suited to the needs of some specified task. Adhering to the assumption that the number of human operators required for the operation of such constellations is linearly correlated with the number of satellites in it [27], this human-operated approach may rapidly become cumbersome. Especially under emergency circumstances where quick decision-making is of great importance, having the selection decisions made by reliable automated methods could be much more effective than trusting it with human agents. Imagine the scenario of a sudden forest fire, which creates the need for the quick selection of an optimal set of satellites to image the affected site. In such a scenario, automation may mean the difference between a quick intervention and a belated one. Hence, one motivating factor behind this work is to automate the process of selection by developing submodular optimization algorithms that efficiently select suitable subsets autonomously, with provable guarantees demonstrating that the produced solution will be trustworthy, robust, and replicable.

3 Preliminaries and Background

In this section, we overview a number of relevant mathematical concepts for the ensuing algorithmic and theoretical developments.

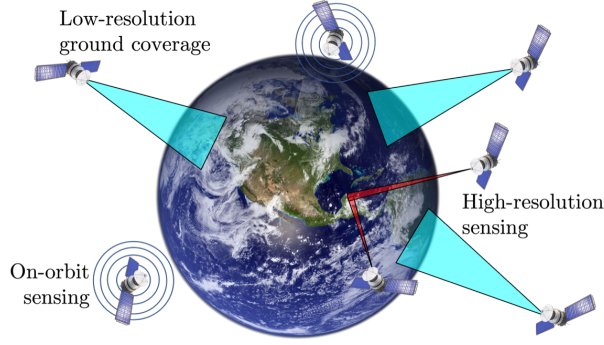


Fig. 1: It is often the case that only some of the satellites in a constellation are actively observing Earth at any given time. Furthermore, these satellites may be leveraged for multi-task missions.

3.1 Notation

We denote the set of nonnegative real numbers by $\mathbb{R}_{\geq 0}$, finite sets using calligraphic notation $\mathcal{S}$, and the n -dimensional identity matrix by I_n . For a finite set $\mathcal{S}$, we denote its cardinality by $|\mathcal{S}|$ and its power set by $2^{\mathcal{S}}$. Finally, we compactly express $[n] := \{1, \dots, n\}$.

3.2 Submodular Maximization

In this section, we provide an overview of submodular maximization. We begin with the following definitions.

Definition 1 (Monotone nondecreasing set functions). A set function $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ is *monotone nondecreasing* if for every $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{N}$, we have $f(\mathcal{S}) \leq f(\mathcal{T})$.

Definition 2 (Submodularity). A set function $f : 2^{\mathcal{N}} \rightarrow \mathbb{R}$ is submodular if

$$f(\mathcal{S} \cup \{j\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{j\}) - f(\mathcal{T}) \quad (1)$$

for all subsets $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{N}$ and $j \in \mathcal{N} \setminus \mathcal{T}$. We also write $f_j(\mathcal{S}) := f(\mathcal{S} \cup \{j\}) - f(\mathcal{S})$ for the marginal return of adding j to $\mathcal{S}$.

In the practical sense, the function f is a performance objective (e.g., the additive inverse of the mean square error (MSE) [15, 28]) and $\mathcal{N}$ is the set of all sensors, out of which suitable selection should be made, commonly referred to as the *ground set*. In this paper, we consider the setting where each sensor $j \in \mathcal{N}$ has a cost of selection $c_j \in \mathbb{R}_{\geq 0}$, and the total cost of a selection $\mathcal{S}$ is denoted by $c(\mathcal{S}) := \sum_{j \in \mathcal{S}} c_j$. Note that when $c_j = 1$ for all $j \in \mathcal{N}$, we have $c(\mathcal{S}) = |\mathcal{S}|$. Let $\mathcal{C} := \{c_i : i \in [|\mathcal{N}|]\}$ and define the ordered set $\bar{\mathcal{C}} := \{\bar{c}_i \in \mathcal{C} : \bar{c}_i \leq \bar{c}_j \forall j \geq i\}$ such that $\bar{\mathcal{C}} = \mathcal{C}$.

For a monotone nondecreasing submodular function f which is, in addition, normalized (i.e., $f(\emptyset) = 0$), we are interested in solving the following budget-constrained combinatorial optimization problem:

$$\begin{aligned} \max_{\mathcal{S} \subseteq \mathcal{N}} & f(\mathcal{S}) \\ \text{s.t. } & c(\mathcal{S}) \leq B, \end{aligned} \quad (2)$$

where B denotes the budget bound. By a reduction to the well-known set cover problem, (2) is known to be NP-hard [10, 29]. Note that when $c_j = 1$ for all $j \in \mathcal{N}$, (2) reduces to the

cardinality-constrained observation selection problem studied in [15, 28]. In this special case, the simple GREEDY algorithm that iteratively selects the remaining element with the highest marginal gain

$$j^* \in \arg \max_{j \in \mathcal{N} \setminus \mathcal{S}^{(i-1)}} f_j(\mathcal{S}^{(i-1)}), \quad (3)$$

where $\mathcal{S}^{(i-1)}$ is the subset selected after the $(i-1)$ th iteration, satisfies the optimal worst-case approximation ratio $f(\mathcal{S}) \geq (1 - 1/e) f(\mathcal{S}^*)$, in which $\mathcal{S}$ is the subset selected by GREEDY, and $\mathcal{S}^*$ is an optimal solution of (2) [11]. The simple extension of this result to the general, budget-constrained case is done through the MODIFIED GREEDY (MG) algorithm, which instead selects the remaining element with the highest the marginal-gain-to-cost ratio $f_j(\mathcal{S}^{(i-1)})/c_j$ and adds it to the constructed solution as long as this addition does not violate the budget constraint. Reminiscent of the well-known result for the GREEDY algorithm, MG is shown to achieve a $\frac{1}{2}(1 - 1/e)$ worst-case approximation ratio [13].

3.3 Weak Submodularity

In many sensor selection tasks, it is observed that f is not strictly submodular but shows similar behavior to submodularity under certain conditions. We will call such functions *weak submodular*, and capture their violation of submodularity through the quantity we will call the *weak-submodularity constant* [30–32].

Definition 3 (Weak-submodularity constant (WSC)). The WSC of a monotone nondecreasing set function f is defined as

$$w_f := \max_{(\mathcal{S}, \mathcal{T}, j) \in \tilde{\mathcal{N}}} \frac{f_j(\mathcal{T})}{f_j(\mathcal{S})}, \quad (4)$$

where $\tilde{\mathcal{N}} = \{(\mathcal{S}, \mathcal{T}, j) : \mathcal{S} \subseteq \mathcal{T} \subset \mathcal{N}, j \in \mathcal{N} \setminus \mathcal{T}\}$.

Intuitively, the WSC of f can be thought of as a measure of quantifying the maximum violation of the marginal diminishing returns property of submodularity. Note that with this definition, we will call any set function f weak submodular if it is monotone nondecreasing and its WSC is bounded. Further note that by Definition 2, a set function f is submodular if and only if its WSC satisfies $w_f \leq 1$ [33–35]. We will generally assume throughout this work that $w_f \geq 1$ to emphasize that the objective function is typically weak submodular.

For a monotone nondecreasing function with a bounded WSC, we have the following proposition [32]:

Proposition 1. Let $w_f \geq 1$ be the WSC of f , a normalized, monotone nondecreasing set function. Then, for two subsets $\mathcal{S}$ and $\mathcal{T}$ such that $\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{N}$, it holds that

$$f(\mathcal{T}) - f(\mathcal{S}) \leq w_f \sum_{j \in \mathcal{T} \setminus \mathcal{S}} f_j(\mathcal{S}). \quad (5)$$

It is worth noting that the notion of weak submodularity is not yet standardized in literature. Alternative notions of weak submodularity exist, such as those presented in [30, 33–35]. Such notions may simplify the derivation of the approximation bounds depending on the application at hand (see e.g., [31, 32, 36–38]). Using the same notion of weak submodularity as the one in this work, [33] extends the theoretical results of [11] on GREEDY to the case of weak submodular functions, obtaining a worst-case approximation ratio of $f(\mathcal{S}) \geq (1 - e^{-1/w_f}) f(\mathcal{S}^*)$.

Algorithm 1 MODIFIED RANDOMIZED GREEDY (MRG)

1: **Input:** Weak-submodular function f , ground set $\mathcal{N}$, budget bound B , cost function c , sample set sizes r_i .
2: **Output:** Subset $\mathcal{S}_{\text{mrg}} \subseteq \mathcal{N}$ with $c(\mathcal{S}_{\text{mrg}}) \leq B$.
3: Initialize $\mathcal{S}^{(0)} \leftarrow \emptyset$, $\mathcal{X} \leftarrow \mathcal{N}$, $i \leftarrow 0$
4: **while** $\mathcal{X} \neq \emptyset$ **do**
5: Form $\mathcal{R}^{(i)}$ by sampling $\min\{r_i, |\mathcal{X}|\}$ elements from $\mathcal{X}$ uniformly at random.
6: $j_s^i \in \arg \max_{j \in \mathcal{R}^{(i)}} \frac{f_j(\mathcal{S}^{(i)})}{c_j}$
7: **if** $c(\mathcal{S}^{(i)}) + c_{j_s^i} \leq B$ **then**
8: $\mathcal{S}^{(i+1)} \leftarrow \mathcal{S}^{(i)} \cup \{j_s^i\}$
9: $i \leftarrow i + 1$
10: **end if**
11: $\mathcal{X} \leftarrow \mathcal{X} - \{j_s^i\}$
12: **end while**
13: $j_{\max} \in \arg \max_{j \in \mathcal{N}} f(\{j\})$
14: $\mathcal{S}_{\text{mrg}} \leftarrow \arg \max_{\mathcal{S} \in \{\mathcal{S}^{(i)}, \{j_{\max}\}\}} f(\mathcal{S})$
15: **return** $\mathcal{S}_{\text{mrg}}$

4 High-Probability Approximation Factors for Greedy Algorithms with Random Sampling

Greedy algorithms rely on a simple philosophy and are relatively straightforward to implement. In practice, however, it is usually undesirable due to computational cost considerations to use the full GREEDY algorithm which searches over the entire set of elements. To overcome this computational burden, we propose two *randomized* greedy algorithms that, at each iteration, consider only a randomly sampled subset of the remaining elements in the ground set. As briefly explained in Section 1, these two algorithms will be used to tackle a budget-constrained weak-submodular sensor selection task, and its dual formulation, i.e., a performance-constrained weak-submodular sensor selection task. Furthermore, we derive performance guarantees that hold with high probability for both algorithms.

4.1 Budget-Constrained Sensor Selection

We start our high-probability analysis by studying how MRG (Algorithm 1) performs in budget-constrained weak-submodular sensor selection problems. To this end, we first need to address a fundamental question: *how should we choose the size of the sampling sets $\mathcal{R}^{(i)}$ in each iteration?* Note that we cannot directly utilize the strategies employed in the cardinality-constrained setting, i.e., setting $r_i = \frac{|\mathcal{N}|}{K} \log \frac{1}{\epsilon}$ according to [14, 15, 28], or adopt a progressively increasing schedule $r_i = \frac{|\mathcal{N}|}{K-i} \log \frac{1}{\epsilon}$ as outlined in [39, 40], where K is the number of observations to select and $0 < \epsilon < 1$ is a parameter set by the user. To see the barrier in choosing the sampling set size r_i , note that due to the budget constraint, it is not known a priori how many observations MRG will end up selecting (so, K is unknown). At this point, although intuitively r_i should depend on B and each sensor cost c_j , it is not clear which strategy results in a nontrivial approximation bound. Hence, we adopt a constructive approach where we let $r_i = \frac{|\mathcal{N}|}{U} \log \frac{1}{\epsilon}$ for some $U > 0$ and $0 < \epsilon < 1$ where U is determined as part of our analysis.

We then proceed to follow and extend [15, 41], and begin with a simplifying assumption that

views each iteration of MRG as an approximation to MODIFIED GREEDY for the budget-constraint formulation (2), which can be obtained from Algorithm 1 by setting the sample cardinality r_i equal to $|\mathcal{N}|$. Essentially, GREEDY is a special case of MRG in which the entire set of remaining sensors is sampled at each iteration. MRG is no longer able to consider every remaining element in the ground set at each iteration, so there is no guarantee of it successfully adding the remaining sensor whose marginal-gain-to-cost ratio is maximal as GREEDY does. However, one can still view each iteration of MRG as adding an element satisfying

$$\frac{f_{j_{\text{mrg}}^i}(\mathcal{S}^{(i)})}{c_{j_{\text{mrg}}^i}} \geq \eta^{(i+1)} \max_{j \in \mathcal{N} \setminus \mathcal{S}^{(i)}} \frac{f_j(\mathcal{S}^{(i)})}{c_j} = \eta^{(i+1)} \frac{f_{j_g^i}(\mathcal{S}^{(i)})}{c_{j_g^i}}, \quad (6)$$

where the subscripts mrg and g refer to the sensor selected in the MRG and GREEDY iterations, respectively, and $\eta^{(i)} \sim \mathcal{D}$ is a random variable drawn from some distribution $\mathcal{D}$ such that $\eta^{(i)} \in (0, 1]$ for all $i \in \mathbb{N}$. Essentially, the key idea is that each iteration i of MRG selects a sensor whose marginal-gain-to-cost ratio is at least $\eta^{(i+1)}$ times as good as the one that would be selected by GREEDY. In contrast to [15, 41], which consider i.i.d. draws of η s for the cardinality-constrained scenario, we exploit the fact that each iteration of MRG is dependent on the selections made in the previous iterations. We then adopt the relaxed assumption that the stochastic process $\{\eta^{(i)}\}_{i=1}^\infty$ is a martingale [42].

Definition 4 (Martingale). A stochastic process $\{Y_i\}_{i=1}^\infty$ is a martingale with respect to the stochastic process $\{X_i\}_{i=1}^\infty$ if, for all $n \in \mathbb{N}$, $\mathbb{E}[|Y_n|] < \infty$ and $\mathbb{E}[Y_{n+1} \mid X_0, \dots, X_n] = Y_n$.

Our martingale assumption is then formalized as follows:

Assumption 1 (Martingale assumption). The stochastic process $\{\eta^{(i)}\}_{i=1}^\infty$ is a martingale with respect to the stochastic process $\{\mathcal{S}^{(i)}\}_{i=0}^\infty$. That is, for all $i \in \mathbb{N}$, $\mathbb{E}[\eta^{(i+1)} \mid \mathcal{S}^{(0:i)}] = \eta^{(i)}$, where $\mathcal{S}^{(0:i)} := \{\mathcal{S}^{(0)}, \dots, \mathcal{S}^{(i)}\}$ denotes the random collection of subsets obtained by MRG up to iteration i .

One may question the justifiability of adopting such an assumption. An informal yet intuitive characterization of a martingale is that it is a stochastic process that does not demonstrate a predictable drift [43]. In numerical experiments, this is indeed observed to be the case, rendering Assumption 1 reasonable.

In the following, we use the next lemma to provide a concentration bound on the stochastic process $\{\eta^{(i)}\}_{i=1}^\infty$.

Lemma 1 (Azuma-Hoeffding inequality [42]). Let $\{Y_i\}_{i=1}^\infty$ be a martingale with respect to $\{X_i\}_{i=1}^\infty$ such that $Y_1 = \mu$. If there exists a sequence of real numbers β_i such that for all $i \in \mathbb{N}$, $|Y_i - Y_{i-1}| \leq \beta_i$, then for any $\lambda \in \mathbb{R}_{\geq 0}$,

$$\Pr(Y_n - \mu \leq -\lambda) \leq \exp\left(-\frac{2\lambda^2}{\sum_{i=1}^n \beta_i^2}\right). \quad (7)$$

Using Lemma 1 and Assumption 1, we now study the performance of MRG for weak-submodular maximization subject to a budget constraint. We obtain the following theorem:

Theorem 1 (Performance of MRG). Let $\{\eta^{(i)}\}_{i=1}^\infty$ satisfy Assumption 1 and the conditions of Lemma 1, where, for all $i \in \mathbb{N}$, $\mathbb{E}[\eta^{(i)}] \geq \mu$, for some $\mu \in \mathbb{R}_{\geq 0}$. Then, for any confidence parameter $0 < \delta < 1$, MRG constructs a solution $\mathcal{S}_{\text{mrg}}$ to (2) such that

$$\frac{f(\mathcal{S}_{\text{mrg}})}{f(\mathcal{S}^*)} \geq \frac{1 - \exp\left[-\frac{1}{w_f} \left(\mu - \frac{c_{\max}}{B} \sqrt{\frac{U}{2} \log \frac{1}{\delta}}\right)\right]}{2w_f^2}, \quad (8)$$

with probability at least $1 - \delta$, where $\mathcal{S}^*$ is an optimal solution, U is the smallest integer such that $\sum_{j=1}^U \bar{c}_j \geq B$, and $c_{\max} := \bar{c}_{|\mathcal{X}|}$.

Sketch of proof. The complete proof is stated in [1]. The aim is to prove the result assuming that the selection returned by the inner loop of MRG in Algorithm 1 is limited to the subset $\mathcal{S}^{(i_v)}$ constructed up to the point before the condition in line 7 of the algorithm evaluates to false for the first time, since this will ensure that it will hold in the general case because of the monotone nondecreasing nature of f .

The subsequent applications of (4), (5), and (6), the use of induction over the iterations of the algorithm, and the invocation of Lemma 1 allow us to place an upper bound on the ratio $f(\mathcal{S}^{(i_v)})/f(\mathcal{S}^*)$, completing the proof. ■

One may wonder why lines 13 and 14 are present in the algorithm. The practical and intuitive reason is that these lines prevent certain adversarial scenarios where a single-element solution outperforms the greedily-constructed solution. More pragmatically, the full proof of Theorem 1 uses these lines in the establishment of the performance lower-bound.

Theorem 1 establishes a high-probability bound on the worst-case performance of MRG. Through observation of the right-hand side of (8), one can see that the performance lower bound depends on several factors. Of these, U , w_f , $c_{\max}$ and B are dictated by the problem itself and are not affected by the stochasticity of the algorithm. δ is the confidence parameter, which one sets *a posteriori*, to see the performance lower bound guaranteed at a certain probability. Meaning, δ is also independent of the stochasticity of the algorithm. The only parameter which depends on the actual process of the algorithm is μ , which one may think of as a lower bound on the expected values of all $\mathbb{E}[\eta^{(i)}]$ through the course of the algorithm. Even more intuitively, one can think of this value as a lower bound on the expected discrepancy between the performance of the standard GREEDY procedure and MRG. When $c_{\max}\sqrt{U} \ll B$ (e.g., in the cardinality-constrained setting, where the cardinality bound K satisfies $\sqrt{K} \ll K$), the right-hand side of (8) can be closely approximated by $(1 - e^{-\mu/w_f})/2w_f^2$. Additionally, if f is submodular (for which $w_f = 1$) and $\mu = 1$, i.e., when $r_i = |\mathcal{N}|$ for all i , this last result reduces to $\frac{1}{2}(1 - 1/e)$, the approximation factor of GREEDY [13]. It is worth noting that the values of $c_{\max}$, U , and B in equation (8) are either known beforehand or can be efficiently computed before running MRG. In contrast, ω_f and μ can be estimated through simulation. In certain specialized scenarios, such as minimizing the MSE in specific linear or quadratic observation models, upper bounds for ω_f can be found [15, 32].

The proof of Theorem 1 also provides an explanation as to the previous determination of the size of the sets $\mathcal{R}^{(i)}$ at each step of MRG. Considering that in the scenario with a cardinality constraint, the value of the cardinality bound K matches the size of $\mathcal{S}^*$, and given that $|\mathcal{S}^*| \leq U$, we suggest selecting $r_i = \frac{|\mathcal{N}|}{U} \log \frac{1}{\epsilon}$ sensors during each iteration of MRG. The results of r_i on the runtime and the performance of the algorithm in a practical setting are demonstrated in Section 6, in Table 1 and Figure 3.

It might be counterintuitive to the reader that the stochastic process $\eta^{(i)}$ appears to be unused in the algorithm. Put simply, $\eta^{(i)}$ describes a sequence of approximation ratios of the marginal-utility-per-cost performance of the stochastic greedy selection process to the marginal-utility-per-cost performance of the standard greedy selection process. Clearly, this is not a quantity that is meant to be (or *can* be) used in the design of the algorithm, since one would have no access to realizations of this sequence unless one ran a copy of the standard greedy selection process at each iteration, completely annulling the computational efficiency of MRG and essentially ending up twice as costly. Rather, $\eta^{(i)}$, (along with μ) is a natural byproduct resulting from the course of the algorithm, whose values are of no interest to us. The only quality we would like $\eta^{(i)}$ to possess is

Algorithm 2 DUAL RANDOMIZED GREEDY (DRG)

1: **Input:** Weak-submodular function f , ground set $\mathcal{N}$, performance threshold A , cost function c , sample set sizes r_i .
2: **Output:** Subset $\mathcal{S}_{\text{drg}} \subseteq \mathcal{N}$ with $f(\mathcal{S}_{\text{drg}}) \geq A$.
3: Initialize $\mathcal{S}^{(0)} \leftarrow \emptyset$, $\mathcal{X} \leftarrow \mathcal{N}$, $i \leftarrow 0$
4: **while** $f(\mathcal{S}^{(i)}) < A$ **do**
5: Form $\mathcal{R}^{(i)}$ by sampling $\min\{r_i, |\mathcal{X}|\}$ elements from $\mathcal{X}$ uniformly at random.
6: $j_s^i \in \arg \max_{j \in \mathcal{R}^{(i)}} \frac{f_j(\mathcal{S}^{(i)})}{c_j}$
7: $\mathcal{S}^{(i+1)} \leftarrow \mathcal{S}^{(i)} \cup \{j_s^i\}$
8: $i \leftarrow i + 1$
9: $\mathcal{X} \leftarrow \mathcal{X} - \{j_s^i\}$
10: **end while**
11: $\mathcal{S}_{\text{drg}} \leftarrow \mathcal{S}^{(i)}$
12: **return** $\mathcal{S}_{\text{drg}}$

that it be a martingale, a reasonable assumption that appears to hold in relevant practical settings.

Note that for $\delta = \exp(-2/U(\mu B/c_{\max})^2)$, the right-hand side is exactly equal to 0, and for any value of δ less than or equal to this quantity, no guarantee is provided. In fact, for $\delta = \exp(-2/U(\mu B/c_{\max})^2)$, the right-hand side is exactly equal to 0, and for any value of δ less than or equal to this quantity, no guarantee is provided.

4.2 Performance-Constrained Sensor Selection

We now establish a high-probability bound for the performance of DUAL RANDOMIZED GREEDY (DRG) (Algorithm 2) for the dual problem of (2), where we aim to satisfy a performance threshold while minimizing the total cost of the selected sensors. This problem is formally expressed as

$$\begin{aligned} & \min_{\mathcal{S} \subseteq \mathcal{N}} c(\mathcal{S}) \\ & \text{s.t. } f(\mathcal{S}) \geq A, \end{aligned} \tag{9}$$

for some performance threshold $A \in \mathbb{R}_{\geq 0}$. Note that when $A = f(\mathcal{N})$ and f is submodular, (9) corresponds exactly to the well-known submodular set covering problem [12]. In that case, the standard GREEDY algorithm, which again represents a special case of Algorithm 2 in which $r_i = |\mathcal{N}|$, obtains the optimal approximation factor [12], given by

$$\frac{c(\mathcal{S}_g)}{c(\mathcal{S}^*)} \leq 1 + \log \left(\frac{f_j(\mathcal{S}^{(0)})}{f_j(\mathcal{S}^{(L-1)})} \right) \leq 1 + \log \frac{M}{m}, \tag{10}$$

where L denotes the total number of iterations taken by GREEDY until a solution is returned, $M := \max_{j \in \mathcal{N}} f(\{j\})$, and $m := \min_{j \in \mathcal{N} \setminus \mathcal{S}^{(L-1)}} f_j(\mathcal{S}^{(L-1)})$. Under the same observation made in (6), we establish that DRG achieves a similar approximation factor with high probability.

Theorem 2 (Performance of DRG). Let $\{\eta^{(i)}\}_{i=1}^\infty$ satisfy Assumption 1 and the conditions of Lemma 1, where, for all $i \in \mathbb{N}$, $\mathbb{E}[\eta^{(i)}] \geq \mu$, for some $\mu \in \mathbb{R}_{\geq 0}$. Then, for any confidence parameter $0 < \delta < 1$, DRG finds a solution $\mathcal{S}_{\text{drg}}$ to problem (9) such that, with probability at least $1 - \delta$,

$$\frac{c(\mathcal{S}_{\text{drg}})}{c(\mathcal{S}^*)} \leq \frac{w_f}{\mu} \left[1 + (L - 1) \log w_f + \log \frac{M}{m} \right] + \frac{1}{\mu c(\mathcal{S}^*)} \sqrt{\frac{1}{2} \log \frac{1}{\delta} c^2(\mathcal{S}_{\text{drg}})}, \tag{11}$$

where $\mathcal{S}^*$ is an optimal solution, $L \leq |\mathcal{N}|$ is the number of iterations required by DRG until termination, and $c^2(\mathcal{S}_{\text{drg}}) := \sum_{j \in \mathcal{S}_{\text{drg}}} c_j^2$.

Sketch of proof. The complete proof is stated in [1]. Here, the fundamental idea is to consider a relaxation of (9), which brings it into the domain of linear programming. The solutions to this relaxation can then be shown, by application of the weak-duality theorem, to be solutions to the original problem.

Lastly, Lemma 1 is invoked to obtain the high-probability guarantee, completing the proof. ■

Recall that the standard GREEDY algorithm is a special case of DRG in which $r_i = |\mathcal{N}|$ for all i . In this case, we can set $\mu = \delta = 1$. Using the result of Theorem 2, we directly obtain that

$$\frac{c(\mathcal{S}_{\text{drg}})}{c(\mathcal{S}^*)} \leq w_f \left[1 + (L - 1) \log w_f + \log \frac{M}{m} \right], \quad (12)$$

which extends the result of [12] to the general setting of performance-constrained weak-submodular optimization that we have formulated in (9), and generalizes the result established in [44] for the uniform unit-cost case. This last result naturally reduces to (10) for the case that f is a submodular function, for which $w_f = 1$. Much like the performance bound of MRG, the upper bound on the approximation ratio of DRG is dependent on terms that arise from the problem setting as well as those that are due to the stochasticity of the algorithm. In particular, w_f , $c(\mathcal{S}^*)$ and m are parameters that depend on the nature of the problem at hand and are independent of the algorithm. μ and δ serve very similar purposes to those in the case of MRG, and are already discussed thoroughly in that section. M is interesting in that it depends both on the nature of the problem and the objective functions, and also the selections made by the algorithm up to the $(L - 1)$ th step. Finally, L is a parameter that essentially indicates how long the algorithm runs until a solution is found, and depends entirely on the course of the algorithm. Lastly, it's worth noting that the parameter M can also be calculated efficiently before starting the DRG process, whereas the values of L and m are determined during the execution of the algorithm.

5 A Randomized Method for Worst-Case Robustness

In this section, we study the application of the DRG algorithm to the solution of a standard formulation of the robust (weak) submodular maximization problem under the budget constraint proposed in [45]. We assume that we are given a finite family of weak-submodular, monotone nondecreasing, and normalized functions $f^1, f^2, \dots, f^n$.¹ We formulate the robust problem as follows:

$$\begin{aligned} & \max_{\mathcal{S} \subseteq \mathcal{N}} \min_{i \in [n]} f^i(\mathcal{S}) \\ & \text{s.t. } c(\mathcal{S}) \leq B. \end{aligned} \quad (13)$$

We can visualize this problem as such: We have n distinct performance objectives that we want to maximize by selecting the optimal subset $\mathcal{S} \subseteq \mathcal{N}$. Since we expect a satisfactory performance with respect to each of these objectives, we want to maximize the worst-performing one, all the while staying under a given budget constraint $B \in \mathbb{R}_{\geq 0}$. In [45], the SUBMODULAR SATURATION ALGORITHM (SSA) is proposed, which solves a relaxation of (13) in the setting where all of the objectives are submodular and a cardinality constraint is posed. SSA is presented in Algorithm 3.

¹The superscript notation is used to index the family of functions to avoid confusion with the marginal return notation.

Algorithm 3 SUBMODULAR SATURATION ALGORITHM (SSA) [45]

Input: Finite family of monotone nondecreasing submodular functions $f^1, \dots, f^n$, ground set $\mathcal{N}$, cardinality bound $K \leq |\mathcal{N}|$, relaxation parameter $\alpha \geq 1$

Output: Solution set $\mathcal{S}$

```
1:  $k_m \leftarrow 0$ 
2:  $k_M \leftarrow \min_{i \in [n]} f^i(\mathcal{N})$ 
3:  $\mathcal{S} \leftarrow \emptyset$ 
4: while  $k_M - k_m \geq 1/n$  do
5:    $k \leftarrow (k_M + k_m)/2$ 
6:   Define  $\bar{f}_{(k)}(\mathcal{X}) := \frac{1}{n} \sum_{i=1}^n \min\{f^i(\mathcal{X}), k\}$ 
7:    $\hat{\mathcal{S}} \leftarrow \text{GREEDY}(\bar{f}_{(k)}, k)$ 
8:   if  $|\hat{\mathcal{S}}| > \alpha K$  then
9:      $k_M \leftarrow k$ 
10:  else
11:     $k_m \leftarrow k$ 
12:     $\mathcal{S} \leftarrow \hat{\mathcal{S}}$ 
13:  end if
14: end while
15: return  $\mathcal{S}$ 
```

To facilitate the following discussion, we now provide a brief description of SSA. Given a finite family of normalized, monotone nondecreasing submodular functions $f^1, \dots, f^n$, a cardinality bound $K \in \mathbb{N}$, and a relaxation parameter $\alpha \geq 1$, SSA constructs a solution $\mathcal{S} \subseteq \mathcal{N}$ such that $f^i(\mathcal{S}) \geq k$ for a certain $k > 0$ and $|\mathcal{S}| \leq \alpha K$. Hence, it solves a relaxation of (13), where the budget constraint has been reduced to a cardinality constraint and relaxed by a factor of α .

SSA leverages two useful facts. The first is that when f is a submodular function, so is the truncated function given by $f_{(k)}(\mathcal{X}) := \min\{f(\mathcal{X}), k\}$ [45], which we call f truncated at k . The second is that the nonnegative weighted sum (and hence the mean) of submodular functions is again submodular [45]. The algorithm then truncates the input functions $f^1, \dots, f^n$ and averages them, producing the submodular function $\bar{f}_{(k)}$. It adaptively decides on the truncation value k by doing a line search over the feasible values of k , and outputs a solution $\mathcal{S} \subseteq \mathcal{N}$ with $f^i(\mathcal{S}) \geq k$ for all $i \in [n]$, where k is the highest value found by the line search procedure. As can be seen in line 7, SSA makes use of the GREEDY as a subroutine. The aim of this section is twofold: to extend the result given in [45] using GREEDY to one using the computationally cheaper, randomized version of DRG, and to further generalize the result to the case involving weak submodular functions and a more complex budget constraint with nonuniform costs instead of the simpler cardinality-constrained setting.

Now, with regard to SSA, we have the following theoretical guarantee:

Theorem 3 (Performance of SSA [45]). For any cardinality bound $K \in \mathbb{N}$, SSA finds a solution $\hat{\mathcal{S}}$ to a relaxation of (13), where f^i are submodular for all $i \in [n]$, such that

$$\min_{i \in [n]} f^i(\hat{\mathcal{S}}) > \max_{\mathcal{S} \subseteq \mathcal{N}, c(\mathcal{S}) \leq B} \min_{i \in [n]} f^i(\mathcal{S}) \quad (14)$$

and $c(\hat{\mathcal{S}}) \leq \alpha B$, for $\alpha = 1 + \log(\max_{j \in \mathcal{N}} \sum_{i=1}^n f^i(\{j\}))$.

The above result relies on the following lemma concerning the GREEDY algorithm [46]:

Lemma 2. Given a submodular function f over a ground set $\mathcal{N}$, and a feasible performance threshold value k , the GREEDY algorithm produces a solution $\hat{\mathcal{S}}$ such that

$$\frac{|\hat{\mathcal{S}}|}{|\mathcal{S}^*|} \leq 1 + \log \max_{j \in \mathcal{N}} f(\{j\}), \quad (15)$$

where $\mathcal{S}^*$ is an optimal solution (i.e., $\mathcal{S}^* = \operatorname{argmin}\{|\mathcal{S}| : f(\mathcal{S}) \geq k\}$).

With this result, it becomes clear how Theorem 3 provides this guarantee: since GREEDY is guaranteed to return a solution $\hat{\mathcal{S}}$ such that $|\hat{\mathcal{S}}|/|\mathcal{S}^*| \leq 1 + \log \max_{j \in \mathcal{N}} f(\{j\})$, we may set our relaxation parameter α in SSA to exactly this upper bound on the right-hand side. Doing so, we know that if the GREEDY subroutine fails to return a solution for a given performance threshold k , then that value of k must be infeasible. Consequently, it only remains to find the highest k value that *is* feasible, which is achieved by a simple line search procedure. For the line search, the values for the interval to be searched at the current iteration are set in lines 1 and 2 and are updated after each run of the GREEDY subroutine. If the GREEDY subroutine fails to return a solution, we know that the most recent k value provided was infeasible, so we update the higher end of our search interval to that value (line 9). If, on the other hand, the GREEDY subroutine did return a solution, then the most recent k value provided was feasible. This case does not preclude the possibility that there is yet another, higher k value which is feasible, so we continue the line search by updating the lower end of our search interval to this value (line 11).

It may not be readily apparent how giving the average of the truncated $f^1, \dots, f^n$, i.e., $\bar{f}_{(k)}$ as input into the GREEDY subroutine guarantees that the solution $\hat{\mathcal{S}}$ returned by GREEDY satisfies $f^1(\hat{\mathcal{S}}), \dots, f^n(\hat{\mathcal{S}}) \geq k$. It suffices to see that the average of these functions $f^1(\hat{\mathcal{S}}), \dots, f^n(\hat{\mathcal{S}})$ truncated at k is equal to k if and only if $f^1(\hat{\mathcal{S}}), \dots, f^n(\hat{\mathcal{S}}) \geq k$. In other words, even if a single $f^j(\hat{\mathcal{S}}) < k$, this would make its truncated counterpart $f_{(k)}^j(\hat{\mathcal{S}}) < k$, and hence even if for all other $i \neq j$, $f^i(\hat{\mathcal{S}}) \geq k$ holds, their average would not satisfy the performance threshold, making $\bar{f}_{(k)} = \frac{1}{n} \sum_{i=1}^n f_{(k)}^i(\hat{\mathcal{S}}) < k$.

Before directly analyzing the use of DRG instead of GREEDY as a subroutine in SSA, there yet remains a gap to be bridged between the setting of [45] and this work, that is, the consideration of weak-submodular functions instead of proper submodular functions. To derive a result similar to Theorem 3 in the case of weak-submodular functions, we need to demonstrate that averaging and truncation preserve weak submodularity as it preserves submodularity. This is indeed the case, as shown in the following two lemmas.

Lemma 3. Let $f^1, \dots, f^n$ be weak submodular with WSCs $w_{f^1}, \dots, w_{f^n} < \infty$. Any nonnegative weighted sum $\bar{f}$ of $f^1, \dots, f^n$ is also weak submodular, with $w_{\bar{f}} < \infty$.

In particular, note that this result implies that the mean of a finite number of weak-submodular functions is yet again a weak-submodular function. We now show a similar result for the truncation of a weak-submodular function.

Lemma 4. Let f be a weak-submodular function, i.e., $w_f < \infty$. Then, the truncation of f at $k \leq 0$, which is denoted $f_{(k)}(\mathcal{S}) := \min\{f(\mathcal{S}), k\}$, with the convention that if $f_{(k)_i}(\mathcal{T})/f_{(k)_i}(\mathcal{S}) = 0/0$, then $f_{(k)_i}(\mathcal{T})/f_{(k)_i}(\mathcal{S}) = 0$, is a weak-submodular function with $w_{f_{(k)}} < \infty$.

Proof. By our definition of weak submodularity, any set function that is monotone nondecreasing and has a bounded WSC is weak submodular, so we first show that $f_{(k)}$ is monotone nondecreasing. For any pair of sets $\mathcal{S} \subseteq \mathcal{T}$ where $f(\mathcal{S}), f(\mathcal{T}) \leq k$, $f_{(k)}(\mathcal{S}) = f(\mathcal{S}) \leq f(\mathcal{T}) = f_{(k)}(\mathcal{T})$. For any pair

of sets $\mathcal{S} \subset \mathcal{T}$ where $f(\mathcal{S}) \leq k$ and $f(\mathcal{T}) > k$, $f_{(k)}(\mathcal{S}) = f(\mathcal{S}) \leq k = f_{(k)}(\mathcal{T})$. Finally, for any pair of sets $\mathcal{S} \subseteq \mathcal{T}$ where $f(\mathcal{S}), f(\mathcal{T}) > k$, $f_{(k)}(\mathcal{S}) = k = f_{(k)}(\mathcal{T})$. Hence, $f_{(k)}$ is monotone nondecreasing.

Now, let us recall the following definition:

$$w_{f_{(k)}} := \max_{(\mathcal{S}, \mathcal{T}, i) \in \tilde{\mathcal{N}}} f_{(k)_i}(\mathcal{T}) / f_{(k)_i}(\mathcal{S}). \quad (16)$$

Say that this maximum is attained at $(\mathcal{S}_0, \mathcal{T}_0, i_0)$. Due to the monotone nondecreasing nature of $f_{(k)}$, we have that

$$f_{(k)}(\mathcal{S}_0) \leq f_{(k)}(\mathcal{T}_0) \leq f_{(k)}(\mathcal{T}_0 \cup \{i_0\}), \quad (17)$$

and

$$f_{(k)}(\mathcal{S}_0) \leq f_{(k)}(\mathcal{S}_0 \cup \{i_0\}) \leq f_{(k)}(\mathcal{T}_0 \cup \{i_0\}). \quad (18)$$

Let us say that a truncated function $f_{(k)}$ is *saturated* at $\mathcal{S}$ if $f_{(k)}(\mathcal{S}) = k$. In light of these inequalities, the remainder of the proof reduces to checking whether or not $f_{(k)}$ is saturated at each of $\mathcal{S}_0, \mathcal{T}_0, \mathcal{S}_0 \cup \{i_0\}$, and $\mathcal{T}_0 \cup \{i_0\}$. For instance, if $f_{(k)}$ is saturated at $\mathcal{S}_0$, we have, by convention, $w_{f_{(k)}} = 0/0 = 0 < \infty$. If, on the other hand, $f_{(k)}$ is not saturated at $\mathcal{S}_0$ but at $\mathcal{T}_0$, we have, $w_{f_{(k)}} = 0 < \infty$. Investigating the rest of the valid configurations, we obtain

$$w_{f_{(k)}} \in \mathcal{W} := \left\{ 0, \frac{k - f(\mathcal{T}_0)}{k - f(\mathcal{S}_0)}, \frac{k - f(\mathcal{T}_0)}{f_{(k)_{i_0}}(\mathcal{S}_0)}, w_f \right\}. \quad (19)$$

It is important to note that $f_{(k)_{i_0}}(\mathcal{S}_0) = f_{(k)}(\mathcal{S}_0 \cup \{i_0\}) - f_{(k)}(\mathcal{S}_0)$ is obtained only in the case where $f_{(k)}$ is not saturated at $\mathcal{S}_0 \cup \{i_0\}$ (and hence at $\mathcal{S}_0$), which means that $f_{(k)}(\mathcal{S}_0 \cup \{i_0\}) = f(\mathcal{S}_0 \cup \{i_0\})$ and $f_{(k)}(\mathcal{S}_0) = f(\mathcal{S}_0)$. Hence, it must be that $f_{(k)}(\mathcal{S}_0 \cup \{i_0\}) \neq f_{(k)}(\mathcal{S}_0)$, as otherwise, $(\mathcal{S}_0, \mathcal{T}_0, i_0)$ would have been a maximizer of $w_f := \max_{(\mathcal{S}, \mathcal{T}, i) \in \tilde{\mathcal{X}}} f_i(\mathcal{T}) / f_i(\mathcal{S})$, making $w_f = \infty$, violating the premise that $w_f < \infty$. We conclude, then, that every element of $\mathcal{W}$ is a finite value, i.e., $w_{f_{(k)}} < \infty$. $\blacksquare$

With these two results, we are now ready to present RANDOMIZED WEAK SUBMODULAR SATURATION ALGORITHM (RANDOM-WSSA). To summarize, RANDOM-WSSA uses the main body of SSA, with the GREEDY subroutine replaced by its randomized variant DRG, and furthermore, works in the presence of weak-submodular functions. The full algorithm is presented in Algorithm 4.

From the preceding results, it is straightforward to show how the theoretical guarantee for RANDOM-WSSA is established. We formalize this result in the following theorem.

Theorem 4. Suppose the conditions of Theorem 2 hold. Then, for any confidence parameter $0 < \delta < 1$, and for any budget constraint $B \in \mathbb{R}_{\geq 0}$, RANDOM-WSSA finds a solution $\hat{\mathcal{S}}$ to a relaxation of (13), where f^i are weak submodular, i.e., are monotone nondecreasing with WSCs $w_{f^i} < \infty$ for all $i \in [n]$, such that with probability at least $(1 - \delta)^P$,

$$\min_{i \in [n]} f^i(\hat{\mathcal{S}}) > \max_{c(\mathcal{S}) \leq B} \min_{i \in [n]} f^i(\mathcal{S}) \quad (20)$$

and $c(\hat{\mathcal{S}}) \leq \alpha B$, for

$$\alpha = \frac{w_{\bar{f}_{(k)}}}{\mu} \left[1 + (L - 1) \log(w_{\bar{f}_{(k)}}) + \log \frac{M}{m} \right] + \frac{1}{\mu c(\mathcal{S}^*)} \sqrt{\frac{1}{2} \log \frac{1}{\delta} c^2(\hat{\mathcal{S}})}, \quad (21)$$

where $L \leq |\mathcal{N}|$ is the maximum number of iterations required by DRG until termination throughout all iterations of RANDOM-WSSA, P is the number of iterations required by RANDOM-WSSA until termination, $c^2(\mathcal{S}) := \sum_{j \in \mathcal{S}} c_j^2$, $M = \max_{j \in \mathcal{N}, i \in [n]} f^i(\{j\})$, and $m = \min_{j \in \mathcal{N}, i \in [n]} f_j^i(\mathcal{N} \setminus \{j\})$.

Algorithm 4 RANDOMIZED WEAK SUBMODULAR SATURATION ALGORITHM (RANDOM-WSSA)

Input: Finite family of monotone nondecreasing weak-submodular functions $f^1, \dots, f^n$, ground set $\mathcal{N}$, budget bound $B \in \mathbb{R}_{\geq 0}$, cost function c , relaxation parameter $\alpha \geq 1$

Output: Solution set $\mathcal{S}$

```
1:  $k_m \leftarrow 0$ 
2:  $k_M \leftarrow \min_{i \in [n]} f^i(\mathcal{N})$ 
3:  $\mathcal{S} \leftarrow \emptyset$ 
4: while  $k_M - k_m \geq 1/n$  do
5:    $k \leftarrow (k_M + k_m)/2$ 
6:   Define  $\bar{f}_{(k)}(\mathcal{S}) := \frac{1}{n} \sum_{i=1}^n \min\{f^i(\mathcal{S}), k\}$ 
7:    $\hat{\mathcal{S}} \leftarrow \text{DRG}(\bar{f}_{(k)}, k)$ 
8:   if  $c(\hat{\mathcal{S}}) > \alpha B$  then
9:      $k_M \leftarrow k$ 
10:  else
11:     $k_m \leftarrow k$ 
12:     $\mathcal{S} \leftarrow \hat{\mathcal{S}}$ 
13:  end if
14: end while
15: return  $\mathcal{S}$ 
```

Table 1: Average MRG solution time in seconds over the simulation horizon for various r_i and budgets B . Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard GREEDY algorithm.

	$B = 25$	$B = 50$	$B = 75$	$B = 100$
$r_i = 60$	1.52	1.59	1.59	1.58
$r_i = 120$	2.93	2.95	3.03	3.01
$r_i = 180$	3.96	3.96	3.98	3.96
$r_i = 240$	4.33	4.33	4.39	4.34

Proof. The proof follows immediately from Theorem 2, and Lemmas 1, 3, 4. Note that since the DRG subroutine is run P times throughout the execution of RANDOM-WSSA, and that these runs are statistically independent random events, the high-probability guarantee $(1 - \delta)$ is raised to the exponent P , to account for each of the P times that DRG must succeed in returning a solution. ■

To provide concrete values for the setting of α , note that the upper bound in Theorem 2 is given in terms of the quantities $c^2(\hat{\mathcal{S}})$ and $c(\mathcal{S}^*)$. One may upper-bound the first term instead by $c^2(\mathcal{N})$ and lower-bound the second term instead by 1. In case $c(\mathcal{S}^*) < 1$, one must only scale the cost value c_j of every element $j \in \mathcal{N}$ and the budget B by the factor $1/c_{\min}$, where $c_{\min} = \min_{j \in \mathcal{N}} c_j$. Doing so ensures that for all $\mathcal{S} \subseteq \mathcal{N}$, $c(\mathcal{S}) \geq 1$, and in particular, $c(\mathcal{S}^*) \geq 1$, after which we can safely do the lower-bounding by 1. However, in practical settings, one usually does not have to resort to such strategies, as experimental results demonstrate that RANDOM-WSSA very frequently succeeds to construct solutions for the non-relaxed version of the problem, i.e., with $\alpha = 1$.

6 Numerical Examples

We now corroborate our theoretical results on the proposed algorithms for tasks involving sensor selection in the context of LEO satellite constellations. We consider a Walker-Delta constellation

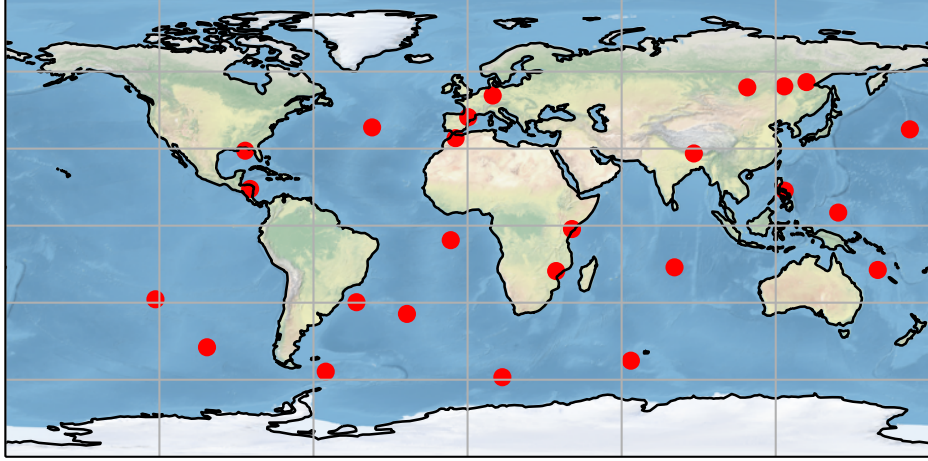


Fig. 2: Locations of the 25 randomly-selected atmospheric points of interest.

Table 2: Average MRG MSE values incurred over the simulation horizon for various r_i and budgets B . Average MSE values for using the entire set and utilizing a Top-K approach are included as benchmarks. Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard GREEDY algorithm.

	$B = 25$	$B = 50$	$B = 75$	$B = 100$
$r_i = 60$	17.73	12.93	10.78	9.52
$r_i = 120$	17.27	12.82	10.78	9.52
$r_i = 180$	17.33	12.83	10.75	9.50
$r_i = 240$	17.23	12.81	10.74	9.51
Entire set	6.89	6.89	6.89	6.89
Top- K	19.39	13.52	11.12	9.74

parameterized by $i : T/P/f$, where i is the orbit inclination, T is the number of satellites, P is the number of unique orbits, and f is the relative spacing between the satellites [47]. The constellation altitude is 2000 km, and the satellites are assumed to remain Earth-pointing with conical field-of-views of angle $\pi/3$.

6.1 Sensor Selection for Atmospheric Monitoring with MRG

Consider a LEO satellite constellation parameterized by $60^\circ : 240/12/1$ performing atmospheric sensing over a set of 25 randomly-instantiated points of interest, as shown in Figure 2. We simulate the conditions at each point $\mathbf{x}_p(t) = [x_p(t), y_p(t), z_p(t)]^\top$ with the Lorenz 63 model [48], a simplified model of atmospheric convection, in which x_p is proportional to the rate of convection while y_p and z_p are proportional to the horizontal and vertical temperature variation, respectively. Precisely, the dynamics at each point are described by

$$\dot{\mathbf{x}}_p = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{z}_p \end{bmatrix} = \kappa \begin{bmatrix} \sigma(y_p - x_p) \\ x_p(\rho - z_p) - y_p \\ x_p y_p - \beta z_p \end{bmatrix} + \omega_p,$$

where $\kappa, \sigma, \rho, \beta \in \mathbb{R}$. Note that we have omitted the dependence on time t for notational clarity. For this simulation, we choose values of $\kappa = 0.005$, $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$. With this selection of parameters, the system exhibits chaotic behavior. Furthermore, $\omega_p \sim \mathcal{N}([0, 0, 0]^\top, \Sigma_{\omega_p})$ is a zero-mean Gaussian process modeling the noise. For this simulation, we set $\Sigma_{\omega_p} = 0.1I_3$.

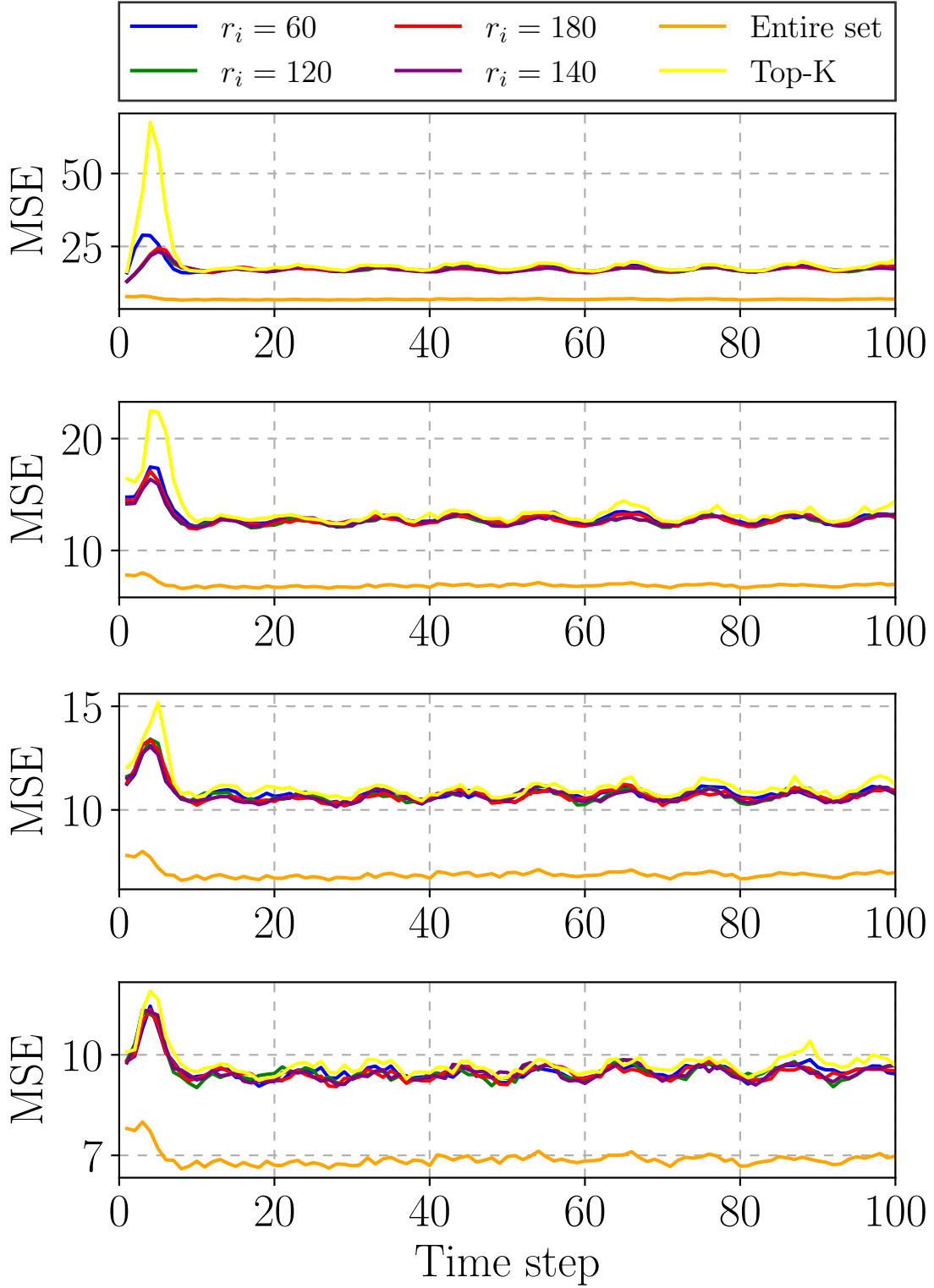


Fig. 3: Total mean-square errors (MSE) over time for various r_i and B . From top to bottom are the plots for $B = 25$, 50, 75, and 100, respectively. Performance of using the entire set of satellites as well as that of picking the highest elements in terms of marginal-gain-to-cost ratio with respect to the empty set until the budget constraint is met (which we call Top- K) are included as benchmark values.

We assign the n th satellite a cost c_n uniformly at random from the interval $[1, 2]$. At each time step of the simulation, the decision-maker aims to make an optimal selection of a subset of the satellites to minimize the MSE of the estimated atmospheric reading, which is known to be a weak-submodular function [28], while staying under a given budget constraint B . In our setting, each time step of the simulation corresponds to $\Delta t = 60$ seconds of time in the simulation. The budget constraint B itself could practically be used to model several real-life limitations, e.g., communication costs between the satellites and the decision-maker or the operational cost of obtaining a reading from a specific satellite. In a real scenario, the cost values of each satellite should be tailored to the specific mission parameters.

The decision-maker uses an unscented Kalman filter [49] to estimate the atmospheric state at each point of interest, using the measurement model

$$\mathbf{z}_{p,n}^s = \mathbf{1}_{p,n}^s \mathbf{x}_p(s\Delta t) + \nu_{p,n}^s, \quad (22)$$

where $\mathbf{z}_{p,n}^s \in \mathbb{R}^3$ is the n^{th} satellite's observation of point p at step s , $\mathbf{1}_{p,n}^s$ is an indicator function equal to 1 if p is in the field-of-view of the n^{th} satellite at step s and 0 otherwise, and $\nu_{p,n}^s \sim \mathcal{N}([0, 0, 0]^\top, \Sigma_{\nu_p})$ is Gaussian measurement noise with $\Sigma_{\nu_p} = 2I_3$. As demonstrated in [28], the objective of minimizing the MSE of the estimator obtained from the selection of satellites can be directly modeled as maximizing a monotone nondecreasing weak-submodular performance objective.

We test several sampling set sizes r_i and budgets B , simulating the estimation task over a horizon of 100 time steps. At each time step, we update the satellite positions and atmospheric conditions. Subsequently, we run MRG at each time step to reconstruct a feasible selection of satellites to use for atmospheric reading. As benchmarks, we include two additional methods of selecting satellites. One consists of not making a selection at all and using the entire constellation of satellites at all times. This demonstrates the ideal scenario of the lowest achievable MSE value at each step, however, it also always incurs the highest cost possible, violating any meaningful constraint on the budget. Additionally, we include the method which we denote by Top- K . This entails sorting the entire set of satellites in decreasing order of marginal-gain-to-cost ratio and then sequentially adding them to the selection as long as they do not violate the budget constraint. Figure 3 shows the resulting MSE throughout the simulation for different combinations of r_i and B values. Table 1 shows the average wall-clock computation time of MRG for each combination. Table 2 shows the average MSE values over the entire time horizon of the simulation incurred by different methods for varying values of r_i and B , effectively displaying the results in Figure 3 in tabulated format. As would be expected, increasing the budget value B allows MRG to add additional satellites whose field-of-view may contain additional points of interest, increasing the total utility achieved, and hence, reducing the incurred MSE. It is interesting to note, however, that although increasing the size r_i of the sampling set at each iteration significantly increases the wall-clock computation time needed to construct solutions, the performance in terms of minimizing the MSE appears to be almost independent of this parameter. This effect is especially pronounced as the budget bound B increases, allowing the randomized selection process more tolerance over the long run. Put another way, as the budget B increases, we implicitly allow the selection process more lenience to make suboptimal selections and then to correct these with better selections with the more ample budget that still remains. This explains the quasi-independence of the performance of MRG from the size of the sampling set r_i , in high-budget scenarios. Recall that the case where $r_i = 240$ corresponds to the standard GREEDY algorithm. It is clear to see the computational advantage of MRG over the full GREEDY algorithm from Table 1. All of these observations demonstrate that, empirically, randomization significantly reduces algorithmic runtime while incurring only a modest

Table 3: Average budget cost incurred by the satellite selection over the simulation horizon for various r_i and coverage fractions F . Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard GREEDY algorithm.

	$F = 0.5$	$F = 0.7$	$F = 0.9$
$r_i = 60$	63.08	99.29	161.74
$r_i = 120$	62.64	99.30	161.47
$r_i = 180$	62.69	99.24	161.34
$r_i = 240$	62.89	99.10	161.24

Table 4: Average DRG solution time in seconds (Algorithm 2) over the simulation horizon for various r_i and coverage fractions F . Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard GREEDY algorithm.

	$F = 0.5$	$F = 0.7$	$F = 0.9$
$r_i = 60$	0.53	0.86	1.37
$r_i = 120$	1.08	1.67	2.64
$r_i = 180$	1.60	2.50	3.64
$r_i = 240$	2.00	2.80	3.96

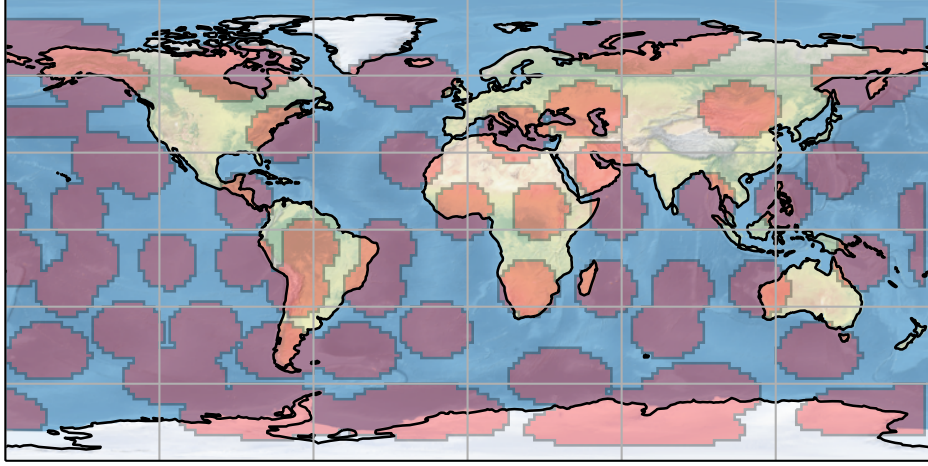
loss in terms of performance. Furthermore, the more sophisticated approach of using a GREEDY algorithm is also justified against using the simpler approach of Top- K , judging by the decrease in MSE values brought about by the introduction of greedy selection. This effect is especially prevalent in lower-budget regimes.

6.2 Minimum Ground Coverage with DRG

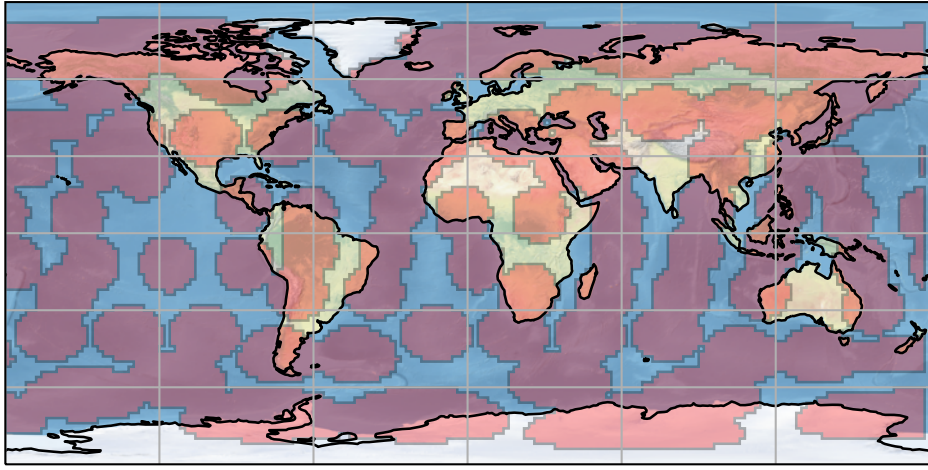
We now consider a scenario where the goal of the decision-maker is to select a subset of satellites whose coverage area is at least a given fraction of the maximum achievable coverage area, which we denote by F while minimizing the incurred cost of the selection. This objective can naturally be expressed in terms of the performance-constrained problem (9). For this simulation, we keep the same constellation parameters as in Section 6.1, with the exception of adjusting the inclination of the constellation to 75° .

To calculate the area covered by a satellite, we first discretize Earth’s surface into a grid of cells of width and height 2° . The area of each cell is then calculated assuming a spherical Earth. At each time step of the simulation, a cell’s area is said to be covered by a satellite if the centroid of that cell happens to be contained inside the field of view of a satellite. Straightforwardly, the marginal return of including a satellite in the selection is the additional area that would be covered by that satellite, that is not already covered by the other satellites in the selection.

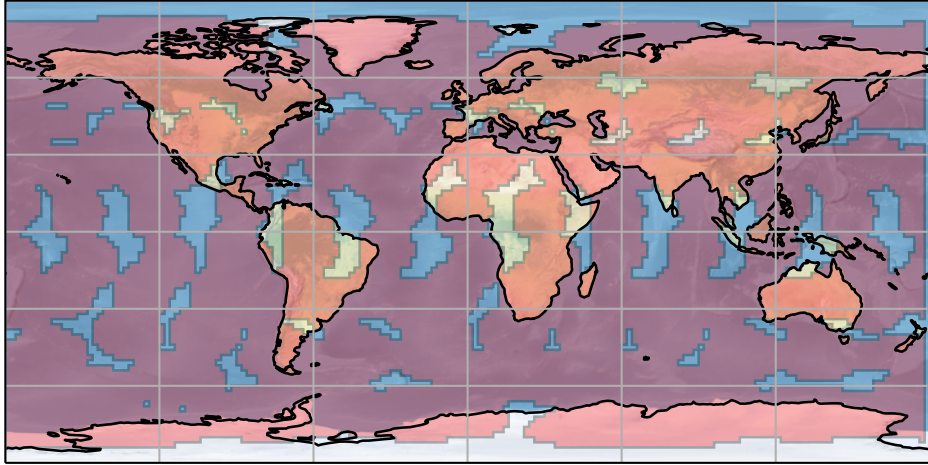
We test varying combinations of r_i and F over a horizon of 100 time steps, at each time step propagating the satellites in their orbit and subsequently running DRG to reconstruct a solution. Figure 4 displays visualizations for the resulting ground coverage obtained using a sampling set size of $r_i = 120$, along with varying F values setting the coverage threshold. Tables 3 and 4 respectively display the average budget cost incurred and the average wall-clock computation time that DRG takes for each combination of r_i and F . We observe from the results that increasing the size of the sampling set in Algorithm 2 generally reduces the average cost of the satellite selection, but only marginally. However, it is clear from Table 4 that increasing the value of r_i causes a significant increase in the wall-clock computation time needed for DRG to produce a solution. The results of both Sections 6.1 and 6.2 demonstrate the benefit of incorporating randomization into the standard GREEDY algorithm, as nearly identical costs for the selection budget can be achieved with a much



(a) Ground coverage obtained for $r_i = 120$ and $F = 0.5$, time step 50.



(b) Ground coverage obtained for $r_i = 120$ and $F = 0.7$, time step 50.



(c) Ground coverage obtained for $r_i = 120$ and $F = 0.9$, time step 50.

Fig. 4: Visualization of the ground coverage areas obtained at time step 50 through DRG using $r_i = 120$ and various F .

Table 5: Average solution time in seconds for RANDOM-WSSA for various r_i and B . Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard SSA.

	$B = 10$	$B = 15$	$B = 20$
$r_i = 15$	39.91	77.56	132.80
$r_i = 30$	84.17	159.74	275.86
$r_i = 45$	128.37	265.39	427.67
$r_i = 240$	647.69	1559.20	2461.96

lower computation time.

6.3 Multi-Objective Robustness with Random-WSSA

Lastly, we move on to experimentally corroborating the results we obtain in Section 5. To reiterate, we deal with a notion of robustness that is ubiquitous in submodular optimization [45, 50–52], wherein we have multiple objectives and aim to produce a solution that performs well with respect to each, subject to a budget constraint.

We instantiate a constellation with the same parameters as in the previous subsection, tasked with accomplishing six distinct tasks. The first five tasks $f^1, \dots, f^5$ are atmospheric sensing tasks as in the experiments of Section 6.1. As opposed to that scenario, each task instantiates five randomly located points on Earth. The utility values $f^1(\mathcal{S}), \dots, f^5(\mathcal{S})$ of these tasks are proportionate to the additive inverse of the mean-squared error (MSE) achieved by the selection $\mathcal{S}$ of satellites.

The sixth task, f^6 , involves ground coverage of the Earth. The utility value $f^6(\mathcal{S})$ of this task is proportionate to the Earth coverage achieved by the selected subset $\mathcal{S}$ of the satellites in the constellation. The area of coverage is determined based on the cell-grid structure explained in Subsection 6.2.

It is important to note that the utility of all six tasks $f^1, \dots, f^6$ is normalized to the range of $[0, 1]$, by dividing the utility $f^i(\mathcal{S})$ of a selection $\mathcal{S}$ on task i at any time step by $f^i(\mathcal{N})$, the maximum utility achievable by selecting the entire ground set at that time step. This ensures that the importance scores of individual tasks are not skewed by the arbitrary value of their utilities.

We run the simulation for 50 time steps and make robust selections using SSA and RANDOM-WSSA with various random sampling set sizes. Figure 5 shows the comparison in terms of the utility achieved by the two algorithms, where the case of $r_i = 240$ corresponds to the standard SSA, and $r_i = 15$, $r_i = 30$ and $r_i = 45$ correspond to running RANDOM-WSSA with various sampling set sizes. The results indicate that RANDOM-WSSA achieves comparable performance to SSA as the size of the sampling set increases, especially in lower-budget settings. However, as is the case with the MRG algorithm, as the budget B increases, we observe that the dependence on r_i weakens, as the increase in budget allows for more room to retroactively correct suboptimal selections. It is also worth noting that both algorithms are run with $\alpha = 1$, i.e., the non-relaxed version of the problem in terms of the budget constraint. Nevertheless, they are consistently able to produce solutions at each time step of the simulation.

Table 5 demonstrates the solution times for RANDOM-WSSA for various combinations of r_i and B . It is clear, in this case, that RANDOM-WSSA enjoys significantly better computational complexity in comparison to SSA, taking nearly twenty times as less time as SSA in some cases, whereas SSA takes longer than 10 minutes on average for a single iteration even in the low-budget scenario of $B = 10$. It is straightforward to conclude, then, that SSA is undesirable in time-critical conditions and that our proposed RANDOM-WSSA is much better-suited to such scenarios.

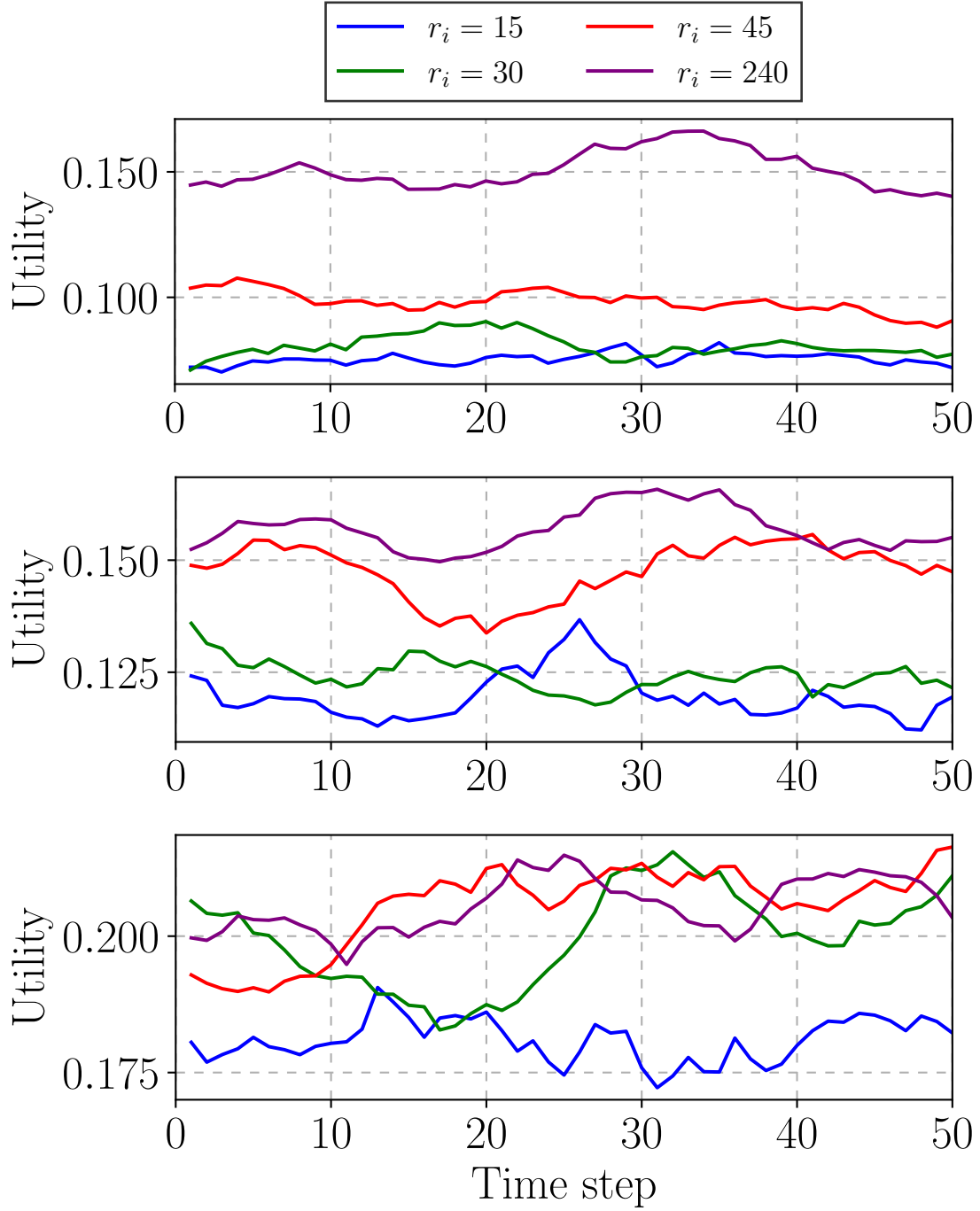


Fig. 5: Total utility over time for various r_i and B for the RANDOM-WSSA. From top to bottom are the plots for $B = 10, 15$, and 20 , and 100 , respectively. The results are put through a moving average filter with a window size of 10 . Note that $r_i = 240$ corresponds to using the entire ground set, i.e., the standard SSA.

7 Conclusion

We introduced three novel algorithms, namely the MODIFIED RANDOMIZED GREEDY (MRG), the DUAL RANDOMIZED GREEDY (DRG), and RANDOMIZED WEAK SUBMODULAR SATURATION ALGORITHM (RANDOM-WSSA). The first two of these algorithms aim to address weak-submodular maximization problems under budget and performance constraints, respectively. The last aims to achieve multi-objective robustness in the presence of performance constraints. Subsequently, we provided theoretical guarantees that hold with high probability on the performance of these algorithms. We illustrated the efficacy of the three algorithms through a practical scenario involving a sensors selection problem, where the sensors in question are satellites in a LEO constellation.

A clear avenue for potential research is the further investigation of the martingale assumption. One could explore whether domain-specific knowledge can enhance the tightness of the obtained bounds, or whether a guarantee that holds on expectation for DRG could be established, with the aim of using it in conjunction with a post-optimization-based method using multiple parallel instances of the algorithm to obtain a guarantee that holds with high probability.

References

- [1] M. Hibbard, A. Hashemi, T. Tanaka, and U. Topcu, “Randomized greedy algorithms for sensor selection in large-scale satellite constellations,” *Proceedings of the American Control Conference*, 2023.
- [2] E. C. Kaya, M. Hibbard, T. Tanaka, U. Topcu, and A. Hashemi, “Randomized greedy methods for weak submodular sensor selection with robustness considerations,” *Automatica*, vol. 171, p. 111984, 2025.
- [3] M. G. Damavandi, V. Krishnamurthy, and J. R. Martí, “Robust meter placement for state estimation in active distribution systems,” *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1972–1982, 2015.
- [4] Z. Liu, A. Clark, P. Lee, L. Bushnell, D. Kirschen, and R. Poovendran, “Towards scalable voltage control in smart grid: a submodular optimization approach,” in *Proceedings of the 7th International Conference on Cyber-Physical Systems*, p. 20, IEEE Press, 2016.
- [5] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, “Submodular maximization with cardinality constraints,” in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’14, (USA), p. 1433–1452, Society for Industrial and Applied Mathematics, 2014.
- [6] S. Dughmi, T. Roughgarden, and M. Sundararajan, “Revenue submodularity,” in *Auctions, Market Mechanisms and Their Applications* (S. Das, M. Ostrovsky, D. Pennock, and B. Szymanski, eds.), (Berlin, Heidelberg), pp. 89–91, Springer Berlin Heidelberg, 2009.
- [7] A. S. Schulz and N. A. Uhan, “Approximating the least core value and least core of cooperative games with supermodular costs,” *Discrete Optimization*, vol. 10, no. 2, pp. 163–180, 2013.
- [8] G. Attigeri, M. Manohara Pai, and R. M. Pai, “Feature selection using submodular approach for financial big data,” *Journal of Information Processing Systems*, vol. 15, no. 6, pp. 1306–1325, 2019.
- [9] R. A. Richards, R. T. Houlette, J. L. Mohammed, *et al.*, “Distributed satellite constellation planning and scheduling,” in *FLAIRS Conference*, pp. 68–72, 2001.
- [10] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge university press, 2011.
- [11] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions,” *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [12] L. A. Wolsey, “An analysis of the greedy algorithm for the submodular set covering problem,” *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.
- [13] S. Khuller, A. Moss, and J. S. Naor, “The budgeted maximum coverage problem,” *Information processing letters*, vol. 70, no. 1, pp. 39–45, 1999.
- [14] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrak, and A. Krause, “Lazier than lazy greedy,” in *AAAI Conference on Artificial Intelligence*, AAAI, 2015.

- [15] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, “Randomized greedy sensor selection: Leveraging weak submodularity,” *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 199–212, 2021.
- [16] A. Krause and D. Golovin, “Submodular function maximization,” in *Tractability: Practical Approaches to Hard Problems*, pp. 71–104, Cambridge University Press, 2014.
- [17] A. Poghosyan and A. Golkar, “CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions,” *Progress in Aerospace Sciences*, vol. 88, pp. 59–83, 2017.
- [18] K. M. Wagner, K. K. Schroeder, and J. T. Black, “Distributed space missions applied to sea surface height monitoring,” *Acta Astronautica*, vol. 178, pp. 634–644, 2021.
- [19] K. Cahoy and A. K. Kennedy, “Initial results from access: an autonomous cubesat constellation scheduling system for earth observation,” 2017.
- [20] W. J. Blackwell, S. Braun, R. Bennartz, C. Velden, M. DeMaria, R. Atlas, J. Dunion, F. Marks, R. Rogers, B. Annane, *et al.*, “An overview of the TROPICS NASA Earth venture mission,” *Quarterly Journal of the Royal Meteorological Society*, vol. 144, pp. 16–26, 2018.
- [21] C. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, “Results from the planet labs flock constellation,” 2014.
- [22] D. Selva and D. Krejci, “A survey and assessment of the capabilities of CubeSats for Earth observation,” *Acta Astronautica*, vol. 74, pp. 50–68, 2012.
- [23] D. Mandl, G. Crum, V. Ly, M. Handy, K. F. Huemrich, L. Ong, B. Holt, and R. Maharaja, “Hyperspectral cubesat constellation for natural hazard response,” in *Annual AIAA/USU conference on small satellites*, no. GSFC-E-DAA-TN33317, 2016.
- [24] G. Santilli, C. Vendittozzi, C. Cappelletti, S. Battistini, and P. Gessini, “CubeSat constellations for disaster management in remote areas,” *Acta Astronautica*, vol. 145, pp. 11–17, 2018.
- [25] S. Nag, J. L. Rios, D. Gerhardt, and C. Pham, “CubeSat constellation design for air traffic monitoring,” *Acta Astronautica*, vol. 128, pp. 180–193, 2016.
- [26] S. Wu, W. Chen, C. Cao, C. Zhang, and Z. Mu, “A multiple-CubeSat constellation for integrated earth observation and marine/air traffic monitoring,” *Advances in Space Research*, vol. 67, no. 11, pp. 3712–3724, 2021.
- [27] S. Siewert and L. McClure, “A system architecture to advance small satellite mission operations autonomy,” 1995.
- [28] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, “A randomized greedy algorithm for near-optimal sensor scheduling in large-scale sensor networks,” in *American Control Conference (ACC)*, pp. 1027–1032, IEEE, 2018.
- [29] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.
- [30] H. Zhang and Y. Vorobeychik, “Submodular optimization with routing constraints,” in *AAAI Conference on Artificial Intelligence*, 2016.

- [31] L. Chamon and A. Ribeiro, “Approximate supermodularity bounds for experimental design,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 5409–5418, 2017.
- [32] A. Hashemi, M. Ghasemi, and H. Vikalo, “Submodular observation selection and information gathering for quadratic models,” in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019.
- [33] A. Das and D. Kempe, “Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1057–1064, 2011.
- [34] E. R. Elenberg, R. Khanna, A. G. Dimakis, and S. Negahban, “Restricted strong convexity implies weak submodularity,” *The Annals of Statistics*, vol. 46, no. 6B, pp. 3539–3568, 2018.
- [35] T. Horel and Y. Singer, “Maximization of approximately submodular functions,” in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3045–3053, 2016.
- [36] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschichatschek, “Guarantees for greedy maximization of non-submodular functions with applications,” in *International Conference on Machine Learning (ICML)*, pp. 498–507, Omnipress, 2017.
- [37] R. Khanna, E. Elenberg, A. Dimakis, S. Negahban, and J. Ghosh, “Scalable greedy feature selection via weak submodularity,” in *Artificial Intelligence and Statistics*, pp. 1560–1568, 2017.
- [38] M. Ghasemi, A. Hashemi, U. Topcu, and H. Vikalo, “On submodularity of quadratic observation selection in constrained networked sensing systems,” in *2019 American Control Conference (ACC)*, pp. 4671–4676, IEEE, 2019.
- [39] A. Hashemi, H. Vikalo, and G. de Veciana, “On the performance-complexity tradeoff in stochastic greedy weak submodular optimization,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3540–3544, IEEE, 2021.
- [40] A. Hashemi, H. Vikalo, and G. de Veciana, “On the benefits of progressively increasing sampling sizes in stochastic greedy weak submodular maximization,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 3978–3992, 2022.
- [41] A. Hassidim and Y. Singer, “Robust guarantees of stochastic greedy algorithms,” in *International Conference on Machine Learning*, pp. 1424–1432, PMLR, 2017.
- [42] F. Chung and L. Lu, “Concentration inequalities and martingale inequalities: a survey,” *Internet Mathematics*, vol. 3, no. 1, pp. 79–127, 2006.
- [43] K. D. Harris, “The martingale z-test,” 2022.
- [44] A. Das and D. Kempe, “Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection,” *Journal of Machine Learning Research*, vol. 19, no. 3, pp. 1–34, 2018.
- [45] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, “Robust submodular observation selection,” *Journal of Machine Learning Research*, vol. 9, no. 93, pp. 2761–2801, 2008.
- [46] L. A. Wolsey, “An analysis of the greedy algorithm for the submodular set covering problem,” *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

- [47] J. G. Walker, “Satellite constellations,” *Journal of the British Interplanetary Society*, vol. 37, p. 559, 1984.
- [48] E. N. Lorenz, “Deterministic nonperiodic flow,” *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.
- [49] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [50] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*, vol. 28. Princeton university press, 2009.
- [51] A. Ben-Tal and A. Nemirovski, “Robust optimization—methodology and applications,” *Mathematical programming*, vol. 92, pp. 453–480, 2002.
- [52] R. Udwani, “Multi-objective maximization of monotone submodular functions with cardinality constraint,” in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.