Recoverable Robust Cardinality Constrained Maximization with Commitment of a Submodular Function[1]

Sabine Münch, Stephen Raach, Sven de Vries

Trier University, 54286 Trier, Germany

{muench/raach/devries}@uni-trier.de

We consider a game-theoretic variant of maximizing a monotone increasing, submodular function under a cardinality constraint. Initially, a solution to this classical problem is determined. Subsequently, a predetermined number of elements from the ground set, not necessarily contained in the initial solution, are deleted, potentially reducing the solution's cardinality. If any deleted elements were part of the initial solution, they are replaced with a set of at most equal cardinality. The objective is to maximize the value of the ultimate solution, with the deletion being maximally disadvantageous to the ultimate solution.

When the submodular function is $M^\natural$-concave, we prove that a simple greedy algorithm computes an optimal solution. When only one element may be deleted, we propose a polynomial running time algorithm with an approximation factor of at least $\frac{1}{3}$. When the number of deletions may become as large as the cardinality parameter, we present a polynomial running time algorithm that approximates an optimal ultimate solution in dependence on the curvature of the submodular function. Furthermore, assuming that the number of allowed deletions is upper bounded by a term of the order of $\frac{k}{\log_2^2(k)}$, where k is the cardinality parameter, we adapt an algorithm from Bogunovic et al. (2017) and show that its approximation factor is at least 0.108.

Keywords : *Recoverable robust optimization, submodular maximization, cardinality constraint, approximation.*

1. Introduction.

Submodular maximization is a fundamental problem in combinatorial optimization, encompassing a variety of classical problems such as graph cuts (see, e.g., Goemans and Williamson 1995), facility location (see, e.g., Ageev and Sviridenko 1999; Cornuejols

[1]This article is an expanded version of Münch et al. (2025), including new results in Section 5, additional explanations, and all proofs omitted in Münch et al. (2025).

et al. 1977), generalized assignment (see, e.g., Feige and Vondrák 2006; Fleischer et al. 2006), and set cover (see, e.g., Krause et al. 2008; Feige 1998). In recent years, submodular maximization has attracted growing attention due to its relevance in many real-world applications of artificial intelligence, which can often be formulated as monotone submodular maximization problems. These include core machine learning tasks such as extractive document and image summarization (see, e.g., Lin and Bilmes 2011; Tschiatschek et al. 2014) and data subset selection (see, e.g., Wei et al. 2015), as well as broader AI-related optimization problems like sensor placement (see, e.g., Krause and Guestrin 2007; Krause et al. 2008) and influence maximization in social networks (see, e.g., Kempe et al. 2015).

Definition 1. *In the following, let I be a finite set of cardinality $n \in \mathbb{N}$ and $f\colon 2^I \to \mathbb{R}_{\geq 0}$ be a* ***submodular*** *($f(X \cup \{i\}) - f(X) \geq f(Y \cup \{i\}) - f(Y)$ for $X \subseteq Y \subseteq I$, $i \in I \setminus Y$),* ***monotone increasing*** *($f(X) \leq f(Y)$ for any $X \subseteq Y \subseteq I$), and* ***normalized*** *($f(\emptyset) = 0$) function.*

We encode the classic problem of *maximizing a submodular function under a cardinality constraint*, denoted as SMC, through instances (I, f, k), where I is a finite set, $f\colon 2^I \to \mathbb{R}_{\geq 0}$ is a monotone increasing, normalized, submodular function, and $\mathbb{N}_{>0} \ni k \leq |I|$ is the cardinality parameter. The objective is to solve

$$\max\{f(X)\colon |X| \leq k,\ X \subseteq I\}.$$

Definition 2. *Let (I, f, k) be an instance of* SMC*. We call a set $S \subseteq I$ with $|S| \leq k$ an* ***optimal solution*** *to (I, f, k) if $f(S) = \max\{f(X)\colon X \subseteq I, |X| \leq k\}$, and denote some arbitrary optimal solution of an instance (I, f, k) as $\mathrm{Opt}^{(I,f,k)}$.*

Recall that the approximation factor of an algorithm is the worst-case ratio between the value obtained by an algorithm and the value of an optimal solution of a problem. For SMC, a simple greedy algorithm, iteratively adding elements that maximize the marginal gain of the objective function until the cardinality parameter is reached, achieves an approximation factor of at least $1 - e^{-1} = 0.63\ldots$, as shown by Nemhauser et al. (1978).

However, in many applications of interest, *robustness* of the solution set against uncertainties about the set of available elements, the cardinality constraint, or the objective function is required, which cannot be guaranteed by the aforementioned greedy algorithm.

Krause et al. (2008) introduced the following *robust version* (RSM) of SMC:

$$\max_{\substack{S \subseteq I \\ |S| \leq k}} \min_{\substack{D \subseteq S \\ |D| \leq w}} f(S \setminus D).$$

The goal of RSM is to determine a solution to a given SMC-instance (I, f, k) such that, after the deletion of at most w elements from the initial solution, the remaining set has an objective function value as high as possible. Hence, an instance of RSM is defined by a tuple (I, f, k, w), where (I, f, k) is an SMC-instance and $1 \leq w \leq k$ is a deletion parameter.

This paper considers a problem closely related to RSM: the *recoverable robust submodular maximization problem under a cardinality constraint with commitment* (abbreviated as RSMC).

Definition 3. *An instance of* RSMC *is defined by a tuple* (I, f, k, w)*, where* (I, f, k) *is an* SMC*-instance and* $1 \leq w \leq k$ *is a* ***deletion parameter****.*

For any RSMC*-instance* (I, f, k, w)*, the* SMC*-instance* (I, f, k) *is called the (unique)* ***underlying instance*** *of* (I, f, k, w)*. Conversely, for any* SMC*-instance* (I, f, k) *and any* $1 \leq w \leq k$*, we call any* RSMC*-instance* (I, f, k, w) *a* ***robust variant*** *of* (I, f, k)*.*

Given an RSMC-instance, a solution to the underlying SMC-instance is computed. Next, a predetermined number of elements, less than or equal to the deletion parameter, is deleted from the ground set. Deleted elements that were part of the computed solution to SMC may be replaced by non-deleted elements.

The deletion is assumed to be adversarial; thus, it is chosen to minimize the value of *our* solution after deletion and replacement.

In contrast, our goal is to select both the initial solution and the replacement elements to maximize the value of the resulting ultimate set; thus, to choose $S \subseteq I$ with $|S| \leq k$, and, after the adversarial choice of $D \subseteq I$ with $|D| \leq w$, to choose $R \subseteq I \setminus (S \cup D)$ with $|R| \leq |S \cap D|$ such that $f((S \setminus D) \cup R)$ approximates

$$\max_{\substack{X \subseteq I \\ |X| \leq k}} \min_{\substack{Y \subseteq I \\ |Y| \leq w}} \max_{\substack{E \subseteq I \setminus (X \cup Y) \\ |E| \leq |X \cap Y|}} f((X \setminus Y) \cup E). \tag{1}$$

In the following, we assume $f(i) > 0$ for all $i \in I$. This assumption is without loss of generality since omitting any chosen element i with $f(i) = 0$ cannot worsen the choice of the selecting party, either the adversary or us.

Definition 4. *Let* (I, f, k, w) *be an* RSMC*-instance. We call any set* $S \subseteq I$ *with* $|S| \leq k$ *an* ***initial solution*** *to* RSMC*, and any set* $D \subseteq I$ *with* $|D| \leq w$ *a* ***deletion set****. For an initial solution* S *and a deletion set* D*, a set* $R(S, D) \subseteq I \setminus (S \cup D)$ *with* $|R(S, D)| \leq |S \cap D|$ *is called a* ***second-stage solution*** *of* S *and* D*. We call* $(S, D, R(S, D))$ *a* ***solution tuple****, and* $(S \setminus D) \cup R(S, D)$ *the corresponding* ***ultimate solution****.*

Given an arbitrary solution tuple $(S, D, R(S, D))$*, we call* $R(S,D)$ *an* ***optimal second-stage solution*** *to* (S, D) *if* $R(S, D) \in \arg\max_{E \subseteq I \setminus (S \cup D), |E| \leq |S \cap D|} f((S \setminus D) \cup E)$*.*

A solution tuple $(S, D, R(S, D))$ *and its corresponding ultimate solution* $(S \setminus D) \cup R(S, D)$ *are called* ***optimal*** *if* $f((S \setminus D) \cup R(S, D))$ *attains the value of* (1)*, which we refer to as* $f(\mathrm{Opt}^{(I,f,k,w)})$*, and if* D *is a minimizer of the ultimate solution given* S *and an optimal second-stage solution; that is,* $D \in \arg\min_{Y \subseteq I, |Y| \leq w} \max_{E \subseteq I \setminus (S \cup Y), |E| \leq |S \cap D|} f((S \setminus Y) \cup E)$*.*

For an optimal solution tuple $(S, D, R(S, D))$*, we call* S *an* ***optimal initial solution****.*

When the initial solution S *and the deletion set* D *are clear from the context, we simplify the notation by writing* (S, D, R) *for* $(S, D, R(S, D))$*.*

We aim to develop a polynomial-time algorithm that, given an instance of RSMC, computes an initial solution and, based on this initial solution and a given deletion set, determines a second-stage solution such that the corresponding ultimate solution approximates an optimal ultimate solution within a constant factor.

Definition 5. *Let $\mathcal{A}$ be an algorithm that computes for any* RSMC*-instance* (I, f, k, w) *an initial solution S, and, given S and an arbitrary but fixed deletion set D, a second stage solution R. Then, the approximation factor of $\mathcal{A}$ is defined as*

$$\min_{\substack{\text{RSMC-}\textit{instance } (I,f,k,w),\\ f(\mathrm{Opt}^{(I,f,k,w)})>0,\\ D\subseteq I,\ |D|\leq w}} \frac{f((S\setminus D)\cup R)}{f(\mathrm{Opt}^{(I,f,k,w)})}.$$

Notice that the approximation factor of an algorithm $\mathcal{A}$ that returns an initial solution and, given this initial solution and a deletion set, a second-stage solution is well-defined, since division by zero is excluded.

1.1. Results.

First, we consider a setting where polynomial-time algorithms can return optimal solutions: For instances (I, f, k, w) with $\mathrm{M}^\natural$-concave objective functions, an important natural subset of submodular functions, we show that the initial and second-stage solution of an optimal solution tuple can be computed in polynomial time, by the greedy algorithm analyzed by Nemhauser et al. (1978). This generalizes a result by Hommelsheim et al. (2023), which focuses on the special case that the $\mathrm{M}^\natural$-concave function is *additionally* the rank function of some matroid.

Next, we examine the special case where the deletion parameter equals one. In this case, we present a polynomial-time algorithm with a constant approximation factor, lying within the interval $\left[\frac{1}{3}, \frac{1-e^{-1}}{2-e^{-1}}\right]$.

Subsequently, we address the general problem and provide a polynomial-time algorithm that achieves an approximation guarantee depending on the curvature of the objective function for any RSMC-instance.

Finally, for RSMC-instances (I, f, k, w) with $w \leq \frac{k}{12\left(\lceil\log_2(\frac{k}{12})\rceil^2+5\lceil\log_2(\frac{k}{12})\rceil\right)}$, based on the results of Bogunovic et al. (2017), we present a polynomial running time algorithm that achieves an approximation factor of at least 0.108. As a byproduct of this result, we demonstrate a new approximation result for RSM. Specifically, we show that this algorithm achieves, for RSM-instances (I, f, k, w) that satisfy the same condition on the number of allowed deletions w, an approximation factor of at least 0.172.

1.2. Related work.

Liebchen et al. (2009) introduced the concept of recoverable robust optimization in the context of timetabling and linear programming for rail optimization. In this framework, the goal is to determine, given an optimization problem with an uncertainty set and a limited set of recovery actions, a feasible solution to the initial optimization problem that can be transformed into a feasible solution in each scenario realization through a recovery action, such that the objective value of the new solution is maximized. For many classical combinatorial problems like shortest path (see, e.g., Büsing 2011; Jackiewicz et al. 2024), knapsack (see, e.g., Büsing et al. 2011b; Büsing et al. 2011a; Büsing et al. 2019), spanning tree (see, e.g., Hradovich et al. 2017a; Hradovich et al. 2017b) and matching (see, e.g., Dourado et al. 2015) complexity has been studied in this framework. For some classical optimization problems, adapted to recoverable robust optimization, even approximation results have been achieved (see, e.g., Lachmann et al. 2021; Goerigk et al. 2022; Bold and Goerigk 2022).

A special case of recoverable robust optimization, called recoverable robust optimization *with commitment*, was studied by Hommelsheim et al. (2023). In this special case, first a feasible solution to an underlying optimization problem defined on a finite ground set of elements is determined. Subsequently, some elements from the ground set are deleted. Then, as a recovery action, it is allowed to add new elements to the remaining initial solution, ensuring that the new solution remains feasible and that all elements from the initial solution that were not deleted remain part of the solution (we commit to the initial solution, or to what remains of it). The objective is to identify an initial solution to the underlying optimization problem, as well as a set of elements to be added post-deletion, to maximize the quality of the resulting solution. Hommelsheim et al. (2023) presented complexity results for several optimization problems adapted to this model. In addition, they showed that an optimal initial solution to the recoverable robust matroid base problem with commitment can be computed by a simple greedy algorithm.

As already mentioned, RSM, a robust version of SMC, was introduced by Krause et al. (2008). Orlin et al. (2018) presented an *asymptotic* constant factor approximation for RSM (with the cardinality parameter growing towards infinity), under the assumption that the number of deletions grows asymptotically as the square root of the cardinality parameter. This result was generalized by Bogunovic et al. (2017) to the assumption that the number of deletions grows asymptotically as the ratio of the cardinality parameter to the cube of its logarithm. RSMC, the problem considered in this paper, is closely related to RSM and a special case of recoverable robust optimization with commitment.

1.3. Brief outline.

In Section 2, we recall the classical greedy algorithm for SMC that starts with an empty solution set and adds iteratively elements according to their marginal gain to the solution set until the cardinality of the solution set matches the cardinality constraint. Furthermore, the connections between optimal solutions to SMC and both the optimal initial and optimal ultimate solutions to RSMC are studied.

In Section 3 and Section 4, we examine special cases of RSMC. Section 3 focuses on RSMC-instances with $M^\natural$-concave objective functions, and Section 4 addresses instances where at most one deletion ($w = 1$) is allowed.

In Section 5, we consider instances where (almost) arbitrarily many deletions are allowed. All omitted proofs are deferred to the Appendix.

2. Maximizing a submodular function under a cardinality constraint.

Before addressing RSMC, we consider the simpler problem of maximizing a submodular function under a cardinality constraint. We recall the classic greedy algorithm for this fundamental problem and provide additional insights that will serve as a basis for addressing the more complex problem of RSMC.

2.1. A greedy algorithm for SMC.

We start by recalling a simple greedy algorithm that gradually chooses the next element according to maximal marginal gain until the cardinality constraint is reached, first analyzed by Nemhauser et al. (1978). To convert this greedy algorithm into a deterministic algorithm, we fix an arbitrary tie-breaking rule for comparing elements by their marginal gain, usually by choosing a permutation. Furthermore, we fix some notation for the marginal gains of elements and sets.

Definition 6. *Let I be a finite set and $f\colon 2^I \to \mathbb{R}$. For any $X, Y \subseteq I$, we define the marginal gain of Y to X by $f(Y|X) \coloneqq f(Y \cup X) - f(X)$. If $Y = \{i\}$ for some $i \in I$, we write $f(i|X)$ for short.*

Algorithm 1:

Input: An instance (I, f, k) of SMC.
Output: Approximately optimal solution.

1 Set $S \leftarrow \emptyset$
2 **while** $|S| \leq k$ **do**
3   Choose $i^* \in \arg\max \{f(i|S) \colon i \in I \setminus S\}$ with tie-breaking according to a fixed rule
4   Set $S \leftarrow S \cup \{i^*\}$
5 **return** S

For our analysis, it is useful to consider the elements of the set S, returned by Algorithm 1, in the order they were added to S. Thus, we assume that iteratively constructed sets are ordered according to the insertion order of all contained elements.

Definition 7. *Let (I, f, k) be an* SMC*-instance, and let $S \subseteq I$ be any set constructed by iteratively adding elements, starting with $S = \emptyset$, e.g., the set constructed in Line 4 of Algorithm 1. In the following, we assume that S is ordered according to the insertion order of the elements, and for any $1 \leq i \leq |S|$, we denote by S_i the element added to S in the i-th iteration.*

We present some auxiliary results concerning the approximation guarantees of Algorithm 1. These will assist in analyzing the algorithms to determine initial and second-stage solutions to RSMC.

We recall the worst-case performance of Algorithm 1.

Proposition 1. *(Nemhauser et al. 1978, Theorem 4.1) For any* SMC*-instance* (I, f, k)*, Algorithm 1 returns a set* S *with* $|S| \leq k$ *such that* $\frac{f(S)}{f(\mathrm{Opt}^{(I,f,k)})} \geq 1 - e^{-1} = 0.63\ldots$ *.*

We show that if the value of the element chosen in the first iteration of the while loop in Line 2 of Algorithm 1 is greater than or equal to half the value of the set S returned by Algorithm 1, then $f(S)$ approximates the value of an optimal solution by at least $\frac{2}{3}$.

Lemma 1. *Let* (I, f, k) *be an* SMC*-instance. Let* S *be the set resulting from Algorithm 1 with* $f(S_1) \geq \frac{f(S)}{2}$*. Then, we have*

$$f(S) \geq \frac{2}{3} f(\mathrm{Opt}^{(I,f,k)}).$$

Proof. By submodularity of f, we have

$$\begin{aligned} f(\mathrm{Opt}^{(I,f,k)}) &\leq f(\{S_1, \ldots, S_{k-1}\}) + \sum_{e \in \mathrm{Opt}^{(I,f,k)} \setminus \{S_1,\ldots,S_{k-1}\}} f(e | \{S_1, \ldots, S_{k-1}\}) \\ &\underset{\text{Line 3 of Alg. 1}}{\leq} f(\{S_1, \ldots, S_{k-1}\}) + \sum_{e \in \mathrm{Opt}^{(I,f,k)} \setminus \{S_1,\ldots,S_{k-1}\}} f(S_k | \{S_1, \ldots, S_{k-1}\}) \\ &\leq f(\{S_1, \ldots, S_{k-1}\}) + k f(S_k | \{S_1, \ldots, S_{k-1}\}) \\ &= f(S) + (k-1) f(S_k | \{S_1, \ldots, S_{k-1}\}. \end{aligned} \tag{2}$$

Furthermore, submodularity and Line 3 of Algorithm 1 imply

$$f(S_k | \{S_1, \ldots, S_{k-1}\}) \leq \frac{f(S) - f(S_1)}{k-1}. \tag{3}$$

By adding $(k-1)$ times Inequality (3) to Inequality (2), we obtain $f(\mathrm{Opt}^{(I,f,k)}) \leq 2f(S) - f(S_1)$. Since $f(S_1) \geq \frac{f(S)}{2}$, it immediately follows that $f(\mathrm{Opt}^{(I,f,k)}) \leq \frac{3}{2} f(S)$. □

We require a classical result by Wolsey (1982) that relates the value of the set returned by Algorithm 1 for an SMC-instance to the optimal solution value of an SMC-instance with the same ground set and objective function, but with a greater cardinality parameter.

Lemma 2. *(Wolsey 1982, Theorem 2) Let* (I, f, k) *be an* SMC*-instance and* S *be the set returned by Algorithm 1. Then,* $f(S) \geq (1 - e^{-\frac{k}{p}}) f(\mathrm{Opt}^{(I,f,p)})$ *for any* $p \geq k$*.*

2.2. Connections between optimal solutions to SMC and RSMC.

We analyze the relationship between optimal solutions to SMC and both the optimal initial and optimal ultimate solutions to RSMC. Note that any solution to an SMC-instance can serve as an initial solution to a robust variant of that instance. However, an optimal solution to SMC is not necessarily an optimal initial solution to a corresponding robust variant.

To provide an example of an initial SMC-solution (together with a deletion set and an optimal second-stage solution) that yields an ultimate solution arbitrarily worse than an optimal ultimate solution, we consider coverage functions, a subset of submodular functions.

Definition 8. *Let $E^1, \dots, E^n$ be a collection of subsets of a finite ground set E. Then, $f\colon 2^{\{1,\dots,n\}} \to \mathbb{R}_{\geq 0}, X \mapsto |\bigcup_{i \in X} E^i|$ is called* ***coverage function****.*

Lemma 3. *Coverage functions are normalized, monotone increasing, and submodular.*

The following example exhibits an RSMC-instance, in which the submodular function is coverage. It demonstrates that for some optimal solution of the underlying SMC-instance as the initial solution, the resulting ultimate solution to the original problem may be arbitrarily worse than the optimal ultimate solution.

Example 1. *For $n \in \mathbb{N}$, let $E = \{1, \dots, n^2 + n - 1\}$, and let $E^1, \dots, E^{2n}$ be a collection of subsets of E with*

$$E^i \coloneqq \begin{cases} \{(i-1)n + 1, \dots, in\} & \text{if } i \in \{1, \dots, n\}, \\ \{n^2 + i - n\} & \text{if } i \in \{n+1, \dots, 2n-1\}, \\ \{1, \dots, n^2\} & \text{if } i = 2n. \end{cases}$$

Let $f^n\colon 2^{\{1,\dots,2n\}} \to \mathbb{R}_{\geq 0}$, $X \mapsto |\bigcup_{i \in X} E^i|$. Then, f^n is a coverage function and hence submodular.

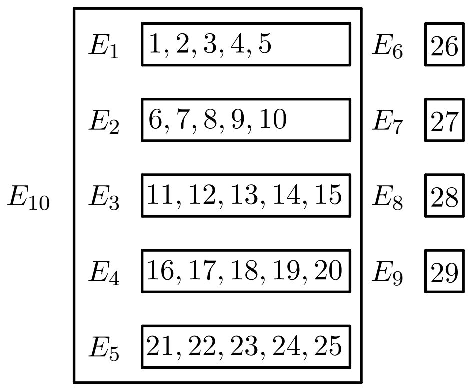

Figure 1: Graphical representation of $E^1, \dots, E^{2n}$ for $n = 5$. Notice that the minimal sets of value 29 are $\{1, 2, \dots, 9\}$ and $\{6, 7, \dots, 10\}$, but only the latter one has cardinality 5.

For $n \in \mathbb{N}$, consider the RSMC-*instance* $(\{1, \dots, 2n\}, f^n, n, 1)$*, with underlying* SMC-*instance* $(\{1, \dots, 2n\}, f^n, n)$*. The set $S = \{n+1, \dots, 2n\}$ is an optimal solution to the* SMC-*instance* (I, f^n, n)*, because $f^n(S) = |\bigcup_{i \in S} E^i| = |\{1, \dots, n^2, n^2 + 1, \dots, n^2 + n - 1\}| = n^2 + n - 1 = f(E)$, and $(S, \{2n\}, \{1\})$ is a solution tuple of $(\{1, \dots, 2n\}, f^n, n, 1)$, where $\{1\}$ is an optimal second-stage solution to the initial solution S and the deletion* *set $\{2n\}$. However, the ultimate solution corresponding to $(S, \{2n\}, \{1\})$ is arbitrarily worse than an optimal ultimate solution since*

$$\lim_{n\to\infty} \frac{f^n((S \setminus \{2n\}) \cup \{1\})}{f^n((\{1,\dots,n\} \setminus \{n\}) \cup \{2n\})} = \lim_{n\to\infty} \frac{2n-1}{n^2} = 0.$$

Although an optimal solution to an SMC-instance is generally not an optimal initial solution of some robust variant, its function value provides a trivial upper bound for the function value of an optimal ultimate solution of any robust variant. To provide a possibly stronger upper bound, we define:

Definition 9. *For any* SMC-*instance* (I, f, k) *and* $T \subseteq I$*, we define the* ***restriction*** *of* (I, f, k) *to* T *as* $(I, f, k)|_T := (T, f|_{2^T}, k)$.

The optimal ultimate solution value of an RSMC-instance (I, f, k, w) is at most the minimum optimal solution value, taken over all subsets $Z \subseteq I$ with $|Z| \leq w$, of the restricted SMC-instances $(I, f, k)|_{I \setminus Z}$.

Theorem 1. *Let* (I, f, k, w) *be an* RSMC-*instance and* (X, D, R) *be an optimal solution tuple to it. Further, let* $Z \in \arg\min_{Y \subseteq I, |Y| \leq w} \max_{E \subseteq I \setminus Y, |E| \leq k} f(E)$*, and let* $T \in \arg\max_{E \subseteq I \setminus Z, |E| \leq k} f(E)$*. Then,*

$$f(T) \geq f((X \setminus D) \cup R).$$

Proof. By the definitions of T and Z, we have

$$\begin{aligned} f(T) &= \max_{E \subseteq I \setminus Z, |E| \leq k} f(E) \geq \max_{E \subseteq I \setminus (Z \cup X), |E| \leq |X \cap Z|} f((X \setminus Z) \cup E) \\ &\geq \min_{Y \subseteq I, |Y| \leq w} \max_{E \subseteq I \setminus (Y \cup X), |E| \leq |X \cap Y|} f((X \setminus Y) \cup E) = f((X \setminus D) \cup R). \end{aligned}$$ □

Notice that Theorem 1 implies that for any RSMC-instance (I, f, k, w) and any $Z \subseteq I$ with $|Z| \leq w$ the function value of the set returned by Algorithm 1 applied to $(I, f, k)|_{I \setminus Z}$ approximates the value of an optimal ultimate solution to (I, f, k, w) by at least $1 - e^{-1}$.

3. RSMC with M$^\natural$-concave objective functions.

Before considering RSMC-instances with general monotone increasing submodular objective functions, we focus on the special case where the objective function is M$^\natural$-concave.

The following definition of M$^\natural$-concavity is due to Murota and Shioura (2018), and is much more convenient than the equivalent original definition by Murota (2003).

Definition 10. *Let* I *be a finite set and* $f\colon 2^I \to \mathbb{R}$*. Then,* f *is called* **M$^\natural$*-concave*** *if for all* $S, T \subseteq I$ *with* $|S| < |T|$*, we have*

$$f(S) + f(T) \leq \max_{t \in T \setminus S} \left\{ f(S \cup \{t\}) + f(T \setminus \{t\}) \right\}.$$

From Definition 10, one can directly observe that the rank functions of matroids form a subset of $\mathrm{M}^\natural$-concave functions.

For RSMC-instances with an $\mathrm{M}^\natural$-concave objective function, we demonstrate that any optimal solution to the underlying SMC-instance is an optimal initial solution. Additionally, we show that an optimal second-stage solution, given an initial solution and a deletion set, can be computed in polynomial running time.

Theorem 2. *Let (I, f, k, w) be an RSMC-instance with $f\colon 2^I \to \mathbb{R}_{\geq 0}$ an $\mathrm{M}^\natural$-concave function. Then, for any optimal solution S to the SMC-instance (I, f, k), there exist D and R such that (S, D, R) is optimal for (I, f, k, w). Furthermore, for any given D, the optimal second-stage solution R can be computed in polynomial running time.*

To prove Theorem 2, we first demonstrate that Algorithm 1 returns an optimal solution for every SMC-instance with an $\mathrm{M}^\natural$-concave objective function.

Lemma 4. *For any SMC-instance (I, f, k) with f $\mathrm{M}^\natural$-concave, Algorithm 1 returns an optimal solution.*

Proof. By Bing et al. (2004, Theorem 2), $\tilde{f}\colon 2^I \to \mathbb{R}_{\geq 0}$, $S \mapsto \max_{T \subseteq S, |T| \leq k} f(T)$ is $\mathrm{M}^\natural$-concave. Furthermore, by Paes Leme (2017, Theorem 3.2), the slight variation of Algorithm 1, which iteratively adds elements maximizing the marginal gain until there are no more elements with a positive marginal gain left, always computes for $\mathrm{M}^\natural$-concave functions an optimal solution. Notice that this version of Algorithm 1 applied to I and $\tilde{f}$ adds at most k elements according to their marginal gain. Therefore, the set returned by this version of Algorithm 1 equals the set returned by Algorithm 1 applied to (I, f, k). Hence, Algorithm 1 is exact. □

The following lemma is equivalent to (Murota 2018, Theorem 2.3), where the term 'valuated matroid' is used to denote the restriction of an $\mathrm{M}^\natural$-concave function to all sets of a fixed cardinality.

Lemma 5. *(Murota 2018, Theorem 2.3.) Let $f\colon 2^I \to \mathbb{R}$ be an $\mathrm{M}^\natural$-concave function. Then, for any $S, T \subseteq I$ with $|S| = |T|$ and any $D \subseteq S \setminus T$, there exists $R \subseteq T \setminus S$, $|R| = |D|$ such that*

$$f(S) + f(T) \leq f((S \setminus D) \cup R) + f((T \setminus R) \cup D).$$

We are ready to prove Theorem 2.

Proof of Theorem 2. Let S be an optimal solution to the SMC-instance (I, f, k) and let D be a deletion set. Further, let $D^S = D \cap S$, and let T be an optimal solution to the SMC-instance $(I, f, k)|_{I \setminus D}$. By Lemma 5, there exists a set $R \subseteq T \setminus S$ with $|R| = |D^S|$ such that

$$f(S) + f(T) \leq f((S \setminus D^S) \cup R) + f((T \setminus R) \cup D^S). \tag{4}$$

By optimality of S and T, we have

$$f((S \setminus D^S) \cup R) \leq f(T) \text{ and } f((T \setminus R) \cup D^S) \leq f(S). \tag{5}$$

Then, Inequalities (4) and (5), and optimality of S and T imply $f((S \setminus D^S) \cup R) = f(T)$ (and $f((T \setminus R) \cup D^S) = f(S)$), and therefore, by Theorem 1, (S, D, R) is an optimal solution tuple.

It remains to show that R can be determined in polynomial running time. Define $\tilde{f}\colon 2^{I \setminus (S \cup D)} \to \mathbb{R}_{\geq 0},\ X \mapsto f((S \setminus D) \cup X)$. Then $\tilde{f}$ is M$^\natural$-concave, since f is M$^\natural$-concave. Let R be the set returned by Algorithm 1 applied to $(I \setminus (S \cup D), \tilde{f}, |S \cap D|)$. Then, R is computed in polynomial running time, and by Lemma 4, we have

$$R \in \underset{E \subseteq I \setminus (S \cup D),\ |E| \leq |S \cap D|}{\arg\max} \tilde{f}(E) = \underset{E \subseteq I \setminus (S \cup D),\ |E| \leq |S \cap D|}{\arg\max} f((S \setminus D) \cup E). \qquad \square$$

Notice that, by Lemma 4 and Theorem 2, for any RSMC-instance with an M$^\natural$-concave objective function, the initial solution and the second-stage solution of an optimal solution tuple can be determined in polynomial running time by using Algorithm 1.

4. RSMC with at most one deletion.

In this section, we focus on instances $(I, f, k, 1)$ of RSMC. As in Section 3, the output of Algorithm 1 for the underlying SMC-instance (I, f, k) is a natural candidate for an initial solution.

Recall that, as shown by Example 1, there exist SMC-instances (I, f, k) with f being a non-M$^\natural$-concave function, such that any optimal solution to (I, f, k), when used as an initial solution to the RSMC-instance $(I, f, k, 1)$, yields an arbitrarily bad ultimate solution. In particular, this may occur when a single element of the optimal solution to SMC contributes the majority of the total value to that solution and, in addition, there does not exist a sufficiently valuable replacement element for it.

The following polynomial-time approximation algorithm avoids this trap by constructing an initial solution such that, for any single element of this kind it contains, either a suitable replacement element exists, which is then returned as the second-stage solution, or the remaining initial solution still retains a sufficiently large value after the element is deleted.

Algorithm 2:

Input: An instance $(I, f, k, 1)$ of RSMC.
Output: Initial solution S and second-stage solution R.

1 Let G be the result of Algorithm 1 applied to (I, f, k)
2 **if** $f(G_1) \leq \frac{f(G)}{2}$ **then**
3   Set $S \leftarrow G$ and $R \leftarrow \emptyset$
4 **else**
5   Let H be the result of Algorithm 1 applied to $(I, f, k)|_{I \setminus \{G_1\}}$
6   Set $S \leftarrow \{H_1, \ldots, H_{k-1}\} \cup \{G_1\}$ and $R \leftarrow \{H_k\}$
7 **return** S, R

Assume that G_1, the first element added by Algorithm 1 to the set G in Line 1 of Algorithm 2, contributes less than or equal to half of the total value of G. Then, Algorithm 2 returns G as an initial solution to the RSMC-instance $(I, f, k, 1)$, since even if the adversary deletes the most valuable element G_1, the remaining initial solution maintains more than half of G's value.

In the case where G_1 is more valuable than half the value of G, Algorithm 2 applies Algorithm 1 to a restriction of the original instance to obtain the set H that is, by itself (in the absence of G_1), a good approximation of an optimal ultimate solution to the RSMC-instance.

Notice that the second-stage solution returned by Algorithm 2 is independent of the deletion set chosen by an adversary; it solely depends on the initial solution.

We provide an approximation factor of at least $\frac{1}{3}$ for Algorithm 2.

Theorem 3. *Algorithm 2 has an approximation factor of at least* $\frac{1}{3}$.

Proof. Let $(I, f, k, 1)$ be an RSMC-instance and let G be the set returned by Algorithm 1 applied to (I, f, k) in Line 1 of Algorithm 2.

Case 1: Assume that Algorithm 2 sets $S = G$ and $R = \emptyset$ in Line 3, and returns these sets in Line 7. Further, let D with $|D| = 1$ be an arbitrary deletion set.

Since Algorithm 1 starts with the empty set and chooses an element with maximum relative gain in every iteration of the while loop, it chooses an element with maximum function value in the first iteration. Hence, $f(G_1) \geq f(G_i)$ and $f(G \setminus \{G_i\}) \geq f(G) - f(G_1)$ for all $G_i \in G$, which directly implies $f(S \setminus D) \geq f(G) - f(G_1)$.

From the proof of Lemma 1, we have $f(\mathrm{Opt}^{(I,f,k)}) \leq 2f(G) - f(G_1)$.

Then,

$$\frac{f((S \setminus D) \cup R)}{f(\mathrm{Opt}^{(I,f,k,1)})} \geq \frac{f(S \setminus D)}{f(\mathrm{Opt}^{(I,f,k)})} \geq \frac{f(G) - f(G_1)}{2f(G) - f(G_1)} = 1 - \frac{f(G)}{2f(G) - f(G_1)} \geq \frac{1}{3},$$

where the last inequality follows by $f(G_1) \leq \frac{f(G)}{2}$.

Case 2: Let H be the set returned by Algorithm 1 applied to $(I, f, k)|_{I \setminus \{G_1\}}$ in Line 5 of Algorithm 2, and assume that Algorithm 2 sets $S = \{H_1, \ldots, H_{k-1}\} \cup \{G_1\}$ and $R = \{H_k\}$ in Line 6 and returns these sets in Line 7.

By Line 3 of Algorithm 2, we have $f(G_1) > \frac{f(G)}{2}$, and by Lemma 1, we have, for all $h \in \{H_1, \ldots, H_{k-1}\}$, that

$$f(S \setminus \{h\}) \geq f(G_1) > \frac{f(G)}{2} \geq \frac{1}{3} f(\mathrm{Opt}^{(I,f,k)}) \geq \frac{1}{3} f(\mathrm{Opt}^{(I,f,k,1)}). \tag{6}$$

Furthermore, it follows, by Proposition 1 and Theorem 1, that

$$\begin{aligned} f((S \setminus \{G_1\}) \cup R) = f(H) &\geq (1 - e^{-1}) f(\mathrm{Opt}^{(I,f,k)|_{I \setminus \{G_1\}}}) \\ &\geq (1 - e^{-1}) f(\mathrm{Opt}^{(I,f,k,1)}). \end{aligned} \tag{7}$$

From (6) and (7), it follows for arbitrary deletion sets D with $|D| = 1$ that $f((S \setminus D) \cup R) \geq \frac{1}{3} f(\mathrm{Opt}^{(I,f,k,1)})$. □

Additionally to Theorem 3, we provide an example demonstrating that the approximation factor of Algorithm 2 is upper bounded by $\frac{1-e^{-1}}{2-e^{-1}} = 0.387\ldots$ in the Appendix. Furthermore, we show that this is the best possible approximation factor achievable when the initial solution is constructed the same way as in Algorithm 2.

5. RSMC with more than one deletion.

Next, we examine RSMC-instances that allow for more than one deletion. Recall that our goal is to construct, for any RSMC-instance, an ultimate solution approximating an optimal ultimate solution by a constant factor in polynomial time. To achieve this, we first aim to determine (given an RSMC-instance (I, f, k, w), an initial solution S, and a deletion set D) a second-stage solution in polynomial time, such that the resulting ultimate solution approximates an optimal ultimate solution by a factor depending only on the number of elements deleted from the initial solution S and the cardinality parameter k. A trivial approach to compute a second-stage solution in polynomial running time is to apply Algorithm 1 to the SMC-instance $(I, f, |S \cap D|)|_{I \setminus D}$ and setting the second-stage solution to the difference of the returned set and S. This approach returns, for any initial solution and any deletion set, in polynomial running time a second-stage solution such that the corresponding ultimate solution approximates an optimal ultimate solution by a factor depending on the ratio $\frac{|S \cap D|}{k}$.

Theorem 4. *Let* (I, f, k, w) *be an* RSMC-*instance and* (S, D, R) *be a solution tuple of* (I, f, k, w)*, where* $R := T \setminus S$ *with* T *being the set returned by Algorithm 1 applied to the* SMC-*instance* $(I, f, |S \cap D|)|_{I \setminus D}$*. Then,*

$$f((S \setminus D) \cup R) \geq (1 - e^{-\frac{|S \cap D|}{k}}) f(\mathrm{Opt}^{(I,f,k,w)}).$$

Proof. By Lemma 2 and Theorem 1, we have

$$f((S \setminus D) \cup R) \geq f(T) \geq (1 - e^{-\frac{|S \cap D|}{k}}) f(\mathrm{Opt}^{(I,f,k)|_{I \setminus D}}) \geq (1 - e^{-\frac{|S \cap D|}{k}}) f(\mathrm{Opt}^{(I,f,k,w)}).$$

□

The lower bound shown in Theorem 4 can become arbitrarily bad if the number of elements deleted from the initial solution is small, relative to the cardinality parameter. Thus, relying solely on the value of an optimal solution to the instance $(I, f, |S \cap D|)|_{I \setminus D}$ of SMC is insufficient to guarantee a good approximation of an optimal ultimate solution; the elements remaining in the initial solution must generally hold significant value.

5.1. A curvature depending approximation algorithm.

To obtain an approximation algorithm for general RSMC-instances, an intuitive approach is to adapt Algorithm 2, which was originally developed for RSMC-instances $(I, f, k, 1)$, to handle arbitrary instances of RSMC. Thus, we consider a natural extension of Algorithm 2 to construct an initial solution to any RSMC-instance (I, f, k, w) and combine it with the previously described method for determining second-stage solutions.

Algorithm 3 consists of two phases. In the first phase, Algorithm 3 determines an initial solution to an RSMC-instance, quite similar to Algorithm 2. In the second phase, Algorithm 3 computes a second-stage solution based on the initial solution and a fixed deletion set D, which is chosen by an adversary and can therefore be viewed as an external input to Algorithm 3.

Algorithm 3:

Input: An RSMC-instance (I, f, k, w).
Output: Initial solution S and second-stage solution R.

Phase 1: Determining an initial solution
Set $S \leftarrow \emptyset$, $S^0 \leftarrow \emptyset$
while $|S^0| < w$ **do**
  Let G be the return of Algorithm 1 applied to $(I, f, k)|_{I \setminus S^0}$
  if $f(\{G_1, \ldots, G_w\}) \leq \frac{f(G)}{2}$ **then**
    Set $S \leftarrow G$
    break
  else
    Set $S^0 \leftarrow S^0 \cup \{G_1\}$
if $S = \emptyset$ **then**
  Let S^1 be the return of Algorithm 1 applied to $(I, f, k - w)|_{I \setminus S^0}$
  Set $S \leftarrow S^0 \cup S^1$
Phase 2: Determining a second-stage solution
Let D be a fixed deletion set chosen by an adversary
Let T be the return of Algorithm 1 applied to $(I, f, |S \cap D|)|_{I \setminus D}$
Set $R \leftarrow T \setminus S$
return S, R

In the following, we demonstrate that Algorithm 3 achieves for each RSMC-instance an approximation factor that is constant except for a multiplicative dependence on the curvature of the objective function of the given instance.

Definition 11. *Let $f: 2^I \to \mathbb{R}_{\geq 0}$ be a normalized, monotone-increasing, and submodular function with $f(i) \neq 0$ for all $i \in I$. The **curvature** of f is defined as*

$$c := 1 - \min_{i \in I} \frac{f(I) - f(I \setminus \{i\})}{f(i)}.$$

Notice that the curvature of a normalized, monotone-increasing, and submodular function f satisfies

$$1 - c \leq \frac{f(S) - f(S \setminus \{i\})}{f(i)} \quad \forall i \in S \subseteq I \text{ and}$$

$$f(S) \geq (1 - c) \sum_{j \in S} f(j).$$

We first demonstrate that an initial solution, returned by Algorithm 3 for some RSMC-instance (I, f, k, w), after deleting any set of cardinality at most w, approximates an optimal ultimate solution depending on the ratio of w and k and the curvature of f.

Lemma 6. *Let (I, f, k, w) be an* RSMC*-instance, and let c be the curvature of f. Furthermore, let S be the initial solution returned by Algorithm 3 applied to (I, f, k, w), and let D be an arbitrary solution set. Then,*

$$f(S \setminus D) \geq \min\left\{(1-c)(1-e^{-(1-\frac{w}{k})}), \frac{1-e^{-1}}{2}\right\} f(\mathrm{Opt}^{(I,f,k,w)}).$$

To prove Lemma 6, we need a simple observation that follows immediately by Lines 3 and 9 of Algorithm 3.

Lemma 7. *Let (I, f, k, w) be an* RSMC*-instance, and S^0 and S^1 be the sets constructed in Lines 9 and 11 of Algorithm 3 applied to (I, f, k, w). Then, $f(i) > f(j)$ for all $i \in S^0$ and all $j \in S^1$.*

Now, we prove Lemma 6.

Proof. Let S^0 and S^1 be the sets constructed in Lines 9 and 11 of Algorithm 3, respectively. We prove Lemma 6 by distinguishing two cases.

Case 1: Assume that Algorithm 3 sets $S = S^0 \cup S^1$ in Line 12. Then, it follows by $|S^1| = k - w$, Lemma 2 and Theorem 1 that

$$f(S^1) \geq (1-e^{-(1-\frac{w}{k})}) f(\mathrm{Opt}^{(I,f,k)|_{I\setminus S^0}}) \geq (1-e^{-(1-\frac{w}{k})}) f(\mathrm{Opt}^{(I,f,k,w)}). \quad (8)$$

Let $D^0 := S^0 \cap D$ and $D^1 := S^1 \cap D$, then we have

$$\begin{aligned}
f(S \setminus D) &= f((S^0 \setminus D^0) \dot\cup (S^1 \setminus D^1)) \underset{\text{Def. 11}}{\geq} (1-c) \sum_{i \in (S^0 \setminus D^0) \dot\cup (S^1 \setminus D^1)} f(i) \\
&\geq (1-c) \left(\sum_{i \in S^0 \setminus D^0} f(i) + \sum_{i \in S^1 \setminus D^1} f(i) \right) \\
&\underset{\text{Lem. 7}}{\geq} (1-c) \left(\frac{|S^0 \setminus D^0|}{|D^1|} \sum_{i \in D^1} f(i) + \sum_{i \in S^1 \setminus D^1} f(i) \right) \\
&= (1-c) \left(\frac{w - |D^0|}{|D^1|} \sum_{i \in D^1} f(i) + \sum_{i \in S^1 \setminus D^1} f(i) \right) \\
&\geq (1-c) \left(\frac{w - (w - |D^1|)}{|D^1|} \sum_{i \in D^1} f(i) + \sum_{i \in S^1 \setminus D^1} f(i) \right) \\
&\geq (1-c) \left(\sum_{i \in D^1} f(i) + \sum_{i \in S^1 \setminus D^1} f(i) \right) \underset{\text{submod. of } f}{\geq} (1-c) f(S^1) \\
&\underset{(8)}{\geq} (1-c)(1-e^{-(1-\frac{w}{k})}) f(\mathrm{Opt}^{(I,f,k,w)}).
\end{aligned}$$

Case 2: Let G be the set returned by Algorithm 1 applied to $(I, f, k)|_{I \setminus S^0}$ in Line 4 of Algorithm 3. Assume that Algorithm 3 sets $S = G$ in Line 6. Let $D^S = D \cap S$ be the set of elements deleted from S. Assume that $D^S = \{D_1^S, \dots, D_{|D^S|}^S\}$ is ordered according to the insertion order of the elements to G by Algorithm 1, i.e., if D_i^S with $i \in \{1, \dots, |D^S|\}$ was added to G in an earlier iteration than D_j^S with $j \in \{1, \dots, |D^S|\}$ then $i \leq j$. Furthermore, for any $l \in \{1, \dots, k\}$ let $S_{<S_l} := \{S_1, \dots, S_{l-1}\} = \{G_1, \dots, G_{l-1}\}$ be the set of all elements added to G in an earlier iteration of Algorithm 1 than the element $G_l = S_l$. Then it follows

$$\begin{aligned}
f(S \setminus D) &= f(S \setminus D^S) = f(S) - (f(S) - f(S \setminus D^S)) \\
&= f(S) - f(D^S | S \setminus D^S) \\
&\geq f(S) - \sum_{j=1}^{|D^S|} f(D_j^S | (S \setminus D^S) \cup \bigcup_{m=1}^{j-1} \{D_m^S\}) \\
&\underset{\text{Line 3 of Alg. 1}}{\geq} f(S) - \sum_{i \in \{S_1, \dots, S_{|D_S|}\}} f(i | S_{<i}) \\
&\geq f(S) - \sum_{i \in \{S_1, \dots, S_w\}} f(i | S_{<i}) \\
&= f(S) - f(\{S_1, \dots, S_w\}) \geq (1 - \frac{1}{2}) f(S) \geq \frac{1 - e^{-1}}{2} f(\mathrm{Opt}^{(I,f,k,w)}),
\end{aligned}$$

where the second last inequality follows by Line 5 of Algorithm 3, and the last inequality follows by Proposition 1 and Theorem 1.

The claim follows from both cases. □

By assuming $\frac{|S \cap D|}{k} \geq 0.5$, which implies $\frac{w}{k} \geq 0.5$, in Theorem 4 and $\frac{w}{k} \leq 0.5$ in Lemma 6, it follows as a simple corollary that Algorithm 3 achieves an approximation factor depending on the curvature of the objective function:

Corollary 1. *For any* RSMC-*instance* (I, f, k, w)*, Algorithm 3 has an approximation factor of at least*

$$\min \left\{ \frac{1 - e^{-1}}{2}, (1 - c)(1 - e^{-0.5}) \right\}, \qquad (*)$$

where c is the curvature of f.

Unfortunately, the approximation factor (∗) of Algorithm 3 is zero, and thereby useless, if the curvature is 1. Consequently, we aim to develop an approximation algorithm with an approximation factor independent of the objective function's curvature.

5.2. An approximation algorithm for instances with a bounded ratio of deletions and cardinality parameter.

To eliminate the dependency of our previous approximation result on the curvature of the objective function, we adapt a polynomial running time approximation algorithm by Bogunovic et al. (2017), originally designed for RSM, to solve RSMC. Before adapting Bogunovic et al.'s algorithm to our setting, we take a brief look at the known results for RSM.

5.2.1. Summary of known results for RSM.

Orlin et al. (2018) provide a polynomial-time algorithm for RSM-instances (I, f, k, w), with $w \in o(\sqrt{\frac{k}{c(k)}})$, achieving an approximation factor of $\frac{e-1}{2e-1}(1 - \frac{1}{\Theta((c(k)))})$, where $c(k)$, with $\lim_{k\to\infty} c(k) = \infty$, is an input parameter of the algorithm that determines the trade-off between how large w can be and how quickly the approximation factor converges to $\frac{1-e^{-1}}{2-e^{-1}} = 0.387\ldots$. This result raised the question of whether an asymptotic constant factor approximation is possible for arbitrary RSM-instances, thus with $w \in o(k)$. The challenge was taken up by Bogunovic et al. (2017), who provided the following algorithm and claimed that it answers the open question in the affirmative.

Algorithm 4:

Input: An RSM-instance (I, f, k, w), $\eta \in \mathbb{N}_{>0}$, and an algorithm $\mathcal{B}$ with the β-iterative property.

Output: Initial solution to (I, f, k, w).

Set $S^0 \leftarrow \emptyset$ and $S^1 \leftarrow \emptyset$
for $i = 0, \ldots, \lceil \log_2(w) \rceil$ **do**
  for $j = 1, \ldots, \lceil \frac{w}{2^i} \rceil$ **do**
    Let $G^{i,j}$ be the set returned by algorithm $\mathcal{B}$ applied to $(I, f, 2^i\eta)|_{I\setminus S^0}$
    Set $S^0 \leftarrow S^0 \cup G^{i,j}$
Let S^1 be the set returned by algorithm $\mathcal{B}$ applied to $(I, f, k - |S^0|)|_{I\setminus S^0}$
Set $S \leftarrow S^0 \cup S^1$
return S

As part of its input, Algorithm 4 requires an algorithm $\mathcal{B}$ with the β-iterative property:

Definition 12. *Let $\mathcal{B}$ be an algorithm that applies to an* SMC*-instance (I, f, k) and returns an iteratively constructed set $G \subseteq I$ with $|G| = k$. For $\beta \geq 1$, algorithm $\mathcal{B}$ has the β-iterative property if for each $j < k$ the following inequality is valid:*

$$f(G_{j+1}|\{G_1, \ldots, G_j\}) \geq \frac{1}{\beta} \max \{f(G^*|\{G_1, \ldots, G_j\})\colon G^* \in I \setminus \{G_1, \ldots, G_j\}\}.$$

Informally, the β-iterative property is based on the idea that an algorithm selects elements iteratively, ensuring that each chosen element has a marginal gain of at least $\frac{1}{\beta}$ of the maximum possible marginal gain. Note that Algorithm 1 satisfies the β-iterative property for $\beta = 1$.

Bogunovic et al. (2017) analyze the approximation guarantee of Algorithm 4 for RSM. For certain RSM-instances (I, f, k, w), where the number of allowed deletions is upper bounded in terms of the cardinality parameter and the input parameter η of Algorithm 4, they prove that the initial solution returned by Algorithm 4, after removing an arbitrary deletion set, approximates an optimal solution of RSM depending on $k, w, |S^0|$ and η.

Proposition 2. *(Bogunovic et al. 2017, Theorem 4.5) Let (I, f, k, w) be an* RSM-*instance with*

$$2 \le w \le \frac{k}{3\eta(\log_2(k)+2)}, \quad \textit{where } 4(\log_2(k)+1) \le \eta \in \mathbb{N}_{>0}.$$

Further, let $\mathcal{B}$ be an algorithm satisfying the β-iterative property. Then, Algorithm 4 returns an initial solution S such that

$$f(S \setminus D) \ge \frac{\frac{\eta}{5\beta^3\lceil\log_2(w)\rceil+\eta}(1-e^{-\frac{k-|S^0|}{\beta(k-w)}})}{1+\frac{\eta}{5\beta^3\lceil\log_2(w)\rceil+\eta}(1-e^{-\frac{k-|S^0|}{\beta(k-w)}})} \cdot \max_{\substack{X\subseteq I\\|X|\le k}} \min_{\substack{Y\subseteq X\\|Y|\le w}} f(X \setminus Y), \tag{9}$$

where $D \in \arg\min_{E\subseteq S, |E|\le w} f(S \setminus E)$.

Notice that Proposition 2 does not directly imply a constant-factor approximation. This is because the analysis by Bogunovic et al. (2017) does not exclude the possibility that the constant in Inequality (9) approaches zero, as $k - |S_0|$ could approach zero.

Additionally, Bogunovic et al. (2017) provide, for RSM-instances with $w \in o(\frac{k}{\eta\log_2(k)})$ and $\eta \ge \log_2^2(k)$, an asymptotic approximation factor for Algorithm 4.

Proposition 3. *(Bogunovic et al. 2017, Theorem 4.5) Let (I, f, k, w) be an* RSM-*instance with $w \in o(\frac{k}{\eta\log_2(k)})$, where $\eta \ge \log_2^2(k)$. Further, let $\mathcal{B}$ be an algorithm satisfying the β-iterative property. Then, Algorithm 4 returns an initial solution S such that for the limit $k \to \infty$:*

$$f(S \setminus D) \ge \left(\frac{1-e^{-\frac{1}{\beta}}}{2-e^{-\frac{1}{\beta}}} + o(1)\right) \max_{\substack{X\subseteq I\\|X|\le k}} \min_{\substack{Y\subseteq X\\|Y|\le w}} f(X \setminus Y),$$

where $D \in \arg\min_{D\subseteq S, |D|\le w} f(S \setminus D)$.

In particular, Algorithm 4 achieves an asymptotic approximation factor of at least $\frac{1-e^{-1}}{2-e^{-1}} = 0.387\ldots$ if Algorithm 1 is used as the subroutine algorithm $\mathcal{B}$.

From Proposition 3, Bogunovic et al. (2017) directly conclude that an asymptotic constant factor approximation is possible for every RSM-instance (I, f, k, w) with $w \in o(k)$. However, in our opinion, this conclusion is incorrect because not every $w \in o(k)$ satisfies $w \in o(\frac{k}{\eta\log_2(k)})$ with $\eta \ge \log_2^2(k)$; for example, $w = \frac{k}{\log_2^2(k)}$.

5.2.2. Adapting Algorithm 4 to RSMC.

Recall that RSMC-instances are similar to RSM-instances. Consequently, Algorithm 4 can be easily adapted to determine an initial solution, given an RSMC-instance. However, Algorithm 4 does not calculate a second-stage solution. In order to nonetheless determine the approximation factor of Algorithm 4, we set each second-stage solution to the empty set in what follows. Hence, Algorithm 4 has a constant approximation factor when it returns, for each RSMC-instance, an initial solution such that the set remaining from the initial solution after removing any arbitrary deletion set approximates an optimal ultimate solution within a constant factor.

Furthermore, in our analysis of Algorithm 4, we fix the input parameter η as well as the algorithm $\mathcal{B}$ used as a subroutine within Algorithm 4.

Definition 13. *Let (I, f, k, w) be an* RSMC*-instance. In the following, we fix the input parameter η of Algorithm 4 to $\eta \coloneqq 4\lceil \log_2(w) \rceil$. Additionally, we set the subroutine algorithm $\mathcal{B}$ of Algorithm 4 to Algorithm 1.*

We demonstrate that Algorithm 4 achieves a constant approximation factor for RSMC-instances (I, f, k, w) with

$$\frac{w}{k} \leq \frac{1}{12\left(\lceil \log_2(\frac{k}{12}) \rceil^2 + 5\lceil \log_2(\frac{k}{12}) \rceil\right)}. \tag{10}$$

Theorem 5. *For any* RSMC*-instance (I, f, k, w) that satisfies Inequality (10), Algorithm 4 has an approximation factor of at least*

$$\frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})}(1 - e^{-\frac{71}{72}}) = 0.108\ldots.$$

The proof of Theorem 5 relies on the following results due to Orlin et al. (2018) and Bogunovic et al. (2017).

Lemma 8. *(Orlin et al. 2018, in Proof of Theorem 12) Let (I, f, k, w) be an* RSMC*-instance, S be the set returned by Algorithm 4, and S^0 be the set constructed in Line 5 of Algorithm 4. Further, let D be a deletion set and $D^0 = S^0 \cap D$. Then,*

$$f(S^0 \setminus D^0) \geq f(\mathrm{Opt}^{(I,f,k-w)|_{I\setminus D^0}}) - f(\mathrm{Opt}^{(I,f,k-w)|_{I\setminus S^0}}).$$

Lemma 9. *(Bogunovic et al. 2017, Lemma D.5, Supplementary) Let (I, f, k, w) be an* RSMC*-instance and D be a deletion set. Further, let $G^{i,j(i)}$, with $i \in \{0, \ldots, \lceil \log_2(w) \rceil\}$ and $j(i) \in \left\{1, \ldots, \left\lceil \frac{w}{2^i} \right\rceil\right\}$, be the set constructed in the i-th iteration of the for loop in Line 2 and $j(i)$-th iteration of the for loop in Line 3 of Algorithm 4. Then, for every $i \in \{0, \ldots, \lceil \log_2(w) \rceil\}$ there exists an $j(i) \in \left\{1, \ldots, \left\lceil \frac{w}{2^i} \right\rceil\right\}$ such that*

$$|G^{i,j(i)} \cap D| \leq 2^i. \tag{11}$$

The following Lemma is a tailored adaptation of Bogunovic et al. (2017, Inequality (66), Supplementary).

Lemma 10. *Let (I, f, k, w) be an* RSMC*-instance, S be the set returned by Algorithm 4, and D a deletion set. Further, let S^1 be the set determined in Line 6 of Algorithm 4, $D^1 = S^1 \cap D$, and for $i \in \{0, \ldots, \lceil \log_2(w) \rceil\}$ and $j(i) \in \{1, \ldots, \lceil \frac{w}{2^i} \rceil\}$ let $G^{i,j(i)}$ be a set constructed in Line 4 of Algorithm 4 that satisfies Inequality (11). Then,*

$$\frac{19}{16} f\Big(\bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D)\Big) \geq f\Big(D^1 \Big| \bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D) \cup (S^1 \setminus D)\Big).$$

Now, the proof of Theorem 5 follows quite similarly to the proof of Bogunovic et al. (2017, Theorem 4.5).

Proof of Theorem 5. Let (I, f, k, w) be an RSMC-instance that satisfies Inequality (10). Further, let S be the set returned by Algorithm 4 and D a deletion set.

To prove Theorem 5, we demonstrate that

$$\frac{f(S \setminus D)}{f(\mathrm{Opt}^{(I,f,k,w)})} \geq \frac{\frac{16}{35}(1-e^{-\frac{2}{3}})}{1+\frac{16}{35}(1-e^{-1})}(1-e^{-\frac{71}{72}}) = 0.108\ldots. \tag{12}$$

Let S^0 be the set in Line 5 and S^1 the set in Line 6 of Algorithm 4.

First, observe that, by the construction of S^0, we have

$$\begin{aligned}
|S^0| &= \sum_{i=0}^{\lceil \log_2(w) \rceil} \left\lceil \frac{w}{2^i} \right\rceil 2^i \cdot 4\lceil \log_2(w) \rceil \leq 4\lceil \log_2(w) \rceil \sum_{i=0}^{\lceil \log_2(w) \rceil} \left(\frac{w}{2^i} + 1 \right) 2^i \\
&= 4\lceil \log_2(w) \rceil \left(w\left(\lceil \log_2(w) \rceil + 1\right) + \sum_{i=0}^{\lceil \log_2(w) \rceil} 2^i \right) \\
&= 4\lceil \log_2(w) \rceil \left(w\left(\lceil \log_2(w) \rceil + 1\right) + \frac{1 - 2^{\lceil \log_2(w) \rceil + 1}}{1-2} \right) \\
&\leq 4\lceil \log_2(w) \rceil \left(w\lceil \log_2(w) \rceil + 5w - 1 \right) \leq 4w \left(\lceil \log_2(w) \rceil^2 + 5\lceil \log_2(w) \rceil \right) \\
&\leq \frac{4k \left(\lceil \log_2(\frac{k}{12}) \rceil^2 + 5\lceil \log_2(\frac{k}{12}) \rceil \right)}{12 \left(\lceil \log_2(\frac{k}{12}) \rceil^2 + 5\lceil \log_2(\frac{k}{12}) \rceil \right)} = \frac{k}{3},
\end{aligned}$$

where the second last inequality follows from Inequality (10).

Let $D^0 = S^0 \cap D$ and $D^1 = S^1 \cap D$, and let m be a constant such that

$$f(D^1 | S \setminus D) = m f(S^1).$$

Now, we provide three lower bounds on $f(S \setminus D)$, each of which depends either on m or on $f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}})$. These bounds can be combined into a single lower bound on $f(S \setminus D)$ that depends solely on $f(\mathrm{Opt}^{(I,f,k,w)})$.

Bound 1: By $S^0 \cap S^0 = \emptyset$, $D^0 \cap D^1 = \emptyset$, we have

$$\begin{aligned}
f(S \setminus D) &= f(S) + (f(S \setminus D^0) - f(S)) + (f(S \setminus D) - f(S \setminus D^0)) \\
&= f(S^1) + f(S^0 | S^1) + (f(S \setminus D^0) - f(D^0 \cup (S \setminus D^0))) \\
&\quad + (f(S \setminus D) - f(D^1 \cup (S \setminus D))) \\
&= f(S^1) + f(S^0 | S^1) - f(D^0 | S \setminus D^0) + f(D^1 | S \setminus D) \\
&\geq f(S^1) - f(D^1 | S \setminus D) = (1-m) f(S^1),
\end{aligned}$$

where the inequality follows from $f(S^0|S^1) - f(D^0|S \setminus D^0) \geq 0$, since $D^0 \subseteq S^0$ and $S^1 \subseteq S \setminus D^0$, and the last equality follows by the definition of m.

Bound 2: For every $i \in \{0, \ldots, \lceil \log_2(w) \rceil\}$, let $j(i) \in \{1, \ldots, \lceil \frac{w}{2^i} \rceil\}$ be such that the set $G^{i,j(i)}$ constructed in the i-th iteration of the for loop in Line 2 and the $j(i)$-th iteration of the for loop in Line 3 of Algorithm 4 satisfies Inequality (11).

By Lemma 10 and submodularity of f, it follows that

$$\begin{aligned} f(S \setminus D) &\geq f(\bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D)) \\ &\geq \frac{16}{19} f(D^1 | \bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D) \cup (S^1 \setminus D)) \\ &\geq \frac{16}{19} f(D^1 | S \setminus D) = \frac{16}{19} m f(S^1). \end{aligned}$$

Bound 3: By Lemma 8, we have

$$f(S \setminus D) \geq f(S^0 \setminus D^0) \geq f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) - f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}}).$$

By combining these three lower bounds on $f(S \setminus D)$, we obtain

$$\begin{aligned} f(S \setminus D) &\geq \max \left\{ f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) - f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}}),\ (1-m) f(S^1), \frac{16}{19} m f(S^1) \right\} \\ &\geq \max \left\{ f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) - f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}}), \frac{16}{35} f(S^1) \right\} \\ &\underset{\text{Lem. 2}}{\geq} \max \left\{ f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) - f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}}), \right. \\ &\left. \frac{16}{35}(1 - e^{-\frac{k-|S^0|}{k-w}}) f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus S^0}}) \right\} \\ &\geq \frac{\frac{16}{35}(1 - e^{-\frac{k-|S^0|}{k-w}})}{1 + \frac{16}{35}(1 - e^{-\frac{k-|S^0|}{k-w}})} f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) \\ &\geq \frac{\frac{16}{35}(1 - e^{-\frac{k-|S^0|}{k}})}{1 + \frac{16}{35}(1 - e^{-\frac{k-|S^0|}{k-w}})} f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) \\ &\geq \frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})} f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}) \\ &\underset{\text{Lem. 2, Th. 1}}{\geq} \frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})} (1 - e^{-\frac{k-w}{k}}) f(\mathrm{Opt}^{(I,f,k,w)}) \\ &\geq \frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})} (1 - e^{-\frac{71}{72}}) f(\mathrm{Opt}^{(I,f,k,w)}) \geq 0.108 f(\mathrm{Opt}^{(I,f,k,w)}) \end{aligned}$$

where the fourth last inequality follows by $|S^0| \leq \frac{k}{3}$ and $|S^0| \geq w$, and the second last inequality follows by $w \leq \frac{k}{12\left(\lceil \log_2(\frac{k}{12})\rceil^2 + 5\lceil \log_2(\frac{k}{12})\rceil\right)} \leq \frac{k}{72}$. Thus, we have demonstrated the validity of Inequality (12) and thereby proven the claim. □

With Theorem 4, we have now obtained an approximation result for certain RSMC-instances (I, f, k, w) that does not depend on the curvature of the objective function, in contrast to Corollary 1. Unfortunately, this improvement comes with the trade-off of restricting the allowed number of deletions to at most $\frac{k}{12(\lceil \log_2(\frac{k}{12})\rceil^2 + 5\lceil \log_2(\frac{k}{12})\rceil)}$. It remains an open question whether it is possible to obtain a constant factor approximation of an optimal ultimate solution for RSMC-instances (I, f, k, w) where more than $\frac{k}{12(\lceil \log_2(\frac{k}{12})\rceil^2 + 5\lceil \log_2(\frac{k}{12})\rceil)}$ deletions are allowed.

5.3. New approximation results for RSM.

As an byproduct of our analysis of Algorithm 4 for RSMC, we obtain a new approximation result of Algorithm 4 for RSM:

Corollary 2. *For any* RSM-*instance* (I, f, k, w) *that satisfies Inequality* (10)*, Algorithm 4 has an approximation factor of at least*

$$\frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})} = 0.172\dots.$$

To prove Corollary 2, we use the following Lemma due to Orlin et al. (2018). It compares the optimal objective value of an RSM-instance to the optimal objective value of the underlying SMC-instance.

Lemma 11. *(Orlin et al. 2018, Lemma 2) For any* RSM-*instance* (I, f, k, w)*, we have:*

$$\max_{\substack{S \subseteq I \\ |S| \leq k}} \min_{\substack{D \subseteq S \\ |D| \leq w}} f(S \setminus D) \leq f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus Z}}) \leq f(\mathrm{Opt}^{(I,f,k-w)}), \text{ for all } Z \subseteq I,\ |Z| \leq w.$$

Now, Corollary 2 follows directly from the inequality

$$f(S \setminus D) \geq \frac{\frac{16}{35}(1 - e^{-\frac{2}{3}})}{1 + \frac{16}{35}(1 - e^{-1})} f(\mathrm{Opt}^{(I,f,k-w)|_{I \setminus D^0}}),$$

shown in the proof of Theorem 5, combined with Lemma 11.

Since Corollary 2 only requires the RSM-instances to satisfy Inequality (10), we have not only shown that Algorithm 4 provides a non-asymptotic constant approximation factor for RSM but also demonstrated its applicability to more RSM-instances, as illustrated by Bogunovic et al. (2017), since Corollary 2 imposes weaker requirements than Proposition 2.

6. Conclusion.

This paper considered the recoverable robust submodular maximization problem under a cardinality constraint with commitment. Assuming that the objective function of an RSMC-instance is $M^\natural$-concave, we demonstrated that the initial solution and the second-stage solution of an optimal solution tuple can be computed in polynomial running time by using a simple greedy algorithm (Algorithm 1). For instances $(I, f, k, 1)$ of RSMC, we presented an $\frac{1}{3}$-approximation algorithm, running in polynomial time.

Considering arbitrary instances of RSMC, we presented a polynomial running time algorithm with an approximation factor that depends on the curvature of the objective function. To eliminate this curvature dependence, we adapted a polynomial running time algorithm by Bogunovic et al. (2017) to RSMC and demonstrated that it achieves an approximation factor of at least 0.108 for any instance (I, f, k, w) of RSMC with $\frac{w}{k} \leq \frac{1}{12\left(\lceil \log_2(\frac{k}{12}) \rceil^2 + 5\lceil \log_2(\frac{k}{12}) \rceil\right)}$. As a byproduct of this result, we improved Bogunovic et al.'s approximation result for RSM by demonstrating that, for instances of RSM fulfilling the same condition, Algorithm 4 achieves a constant approximation factor of at least 0.172.

It remains an interesting open question whether it is possible to achieve a constant factor approximation for RSMC-instances (I, f, k, w) with $\frac{w}{k} > \frac{1}{12\left(\lceil \log_2(\frac{k}{12}) \rceil^2 + 5\lceil \log_2(\frac{k}{12}) \rceil\right)}$.

Acknowledgements.

The authors thank the DFG for their support within RTG 2126 "Algorithmic Optimization".

References

Ageev, A.A. and M.I. Sviridenko (1999). "An 0.828-approximation algorithm for the uncapacitated facility location problem." In: *Discrete Applied Mathematics* 93.2-3, pp. 149–156. DOI: 10.1016/S0166-218X(99)00103-1.

Bing, Meir, Daniel Lehmann, and Paul Milgrom (2004). "Presentation and structure of substitutes valuations." In: *Proceedings of the 5th ACM Conference on Electronic Commerce*. EC '04. Association for Computing Machinery, pp. 238–239. DOI: 10.1145/988772.988812.

Bogunovic, Ilija, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher (2017). "Robust Submodular Maximization: A Non-Uniform Partitioning Approach." In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 508–516.

Bold, Matthew and Marc Goerigk (2022). "Investigating the recoverable robust single machine scheduling problem under interval uncertainty." In: *Discrete Applied Mathematics* 313, pp. 99–114. DOI: 10.1016/j.dam.2022.02.005.

Büsing, Christina (2011). *Recoverable Robustness in Combinatorial Optimization*. Cuvillier Verlag.

Büsing, Christina, Sebastian Goderbauer, Arie M.C.A. Koster, and Manuel Kutschka (2019). "Formulations and algorithms for the recoverable Γ-robust knapsack problem." In: *EURO Journal on Computational Optimization* 7.1, pp. 15–45. DOI: 10.1007/s13675-018-0107-9.

Büsing, Christina, Arie M.C.A. Koster, and Manuel Kutschka (2011a). "Recoverable robust knapsacks: Γ-scenarios." In: *Proceedings of INOC 2011, International Network Optimization Conference*. Vol. 6701. Lecture Notes in Computer Science. Springer-Verlag, pp. 583–588. DOI: 10.1007/978-3-642-21527-8_65.

– (2011b). "Recoverable Robust Knapsacks: the discrete scenario case." In: *Optimization Letters* 5, pp. 379–392. DOI: 10.1007/s11590-011-0307-1.

Cornuejols, Gerard, Marshall L. Fisher, and George L. Nemhauser (1977). "Exceptional Paper - Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms." In: *Management science* 23.8, pp. 789–810. DOI: 10.1287/mnsc.23.8.789.

Dourado, Mitre Costa, Dirk Meierling, Lucia D. Penso, Dieter Rautenbach, Fabio Protti, and Aline Ribeiro de Almeida (2015). "Robust recoverable perfect matchings." In: *Networks* 66.3, pp. 210–213. DOI: 10.1002/net.21624.

Feige, Uriel (1998). "A threshold of ln n for approximating set cover." In: *Journal of the ACM (JACM)* 45.4, pp. 634–652. DOI: 10.1145/285055.285059.

Feige, Uriel and Jan Vondrák (2006). "Approximation algorithms for allocation problems: Improving the factor of 1-1/e." In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, pp. 667–676. DOI: 10.1109/FOCS.2006.14.

Fleischer, Lisa, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko (2006). "Tight Approximation Algorithms for Maximum General Assignment Problems." In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*. SODA '06. Miami, Florida: Society for Industrial and Applied Mathematics, pp. 611–620.

Goemans, Michel X. and David P. Williamson (1995). "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming." In: *Journal of the ACM (JACM)* 42.6, pp. 1115–1145. DOI: 10.1145/227683.227684.

Goerigk, Marc, Stefan Lendl, and Lasse Wulf (2022). "Recoverable robust representatives selection problems with discrete budgeted uncertainty." In: *European Journal of Operational Research* 303.2, pp. 567–580. DOI: 10.1016/j.ejor.2022.03.001.

Hommelsheim, Felix, Nicole Megow, Komal Muluk, and Britta Peis (2023). "Recoverable Robust Optimization with Commitment." In: *Preprint arXiv:2306.08546*. DOI: 10.48550/arXiv.2306.08546.

Hradovich, Mikita, Adam Kasperski, and Paweł Zieliński (2017a). "Recoverable robust spanning tree problem under interval uncertainty representations." In: *Journal of Combinatorial Optimization* 34, pp. 554–573. DOI: 10.1007/s10878-016-0089-6.

– (2017b). "The recoverable robust spanning tree problem with interval costs is polynomially solvable." In: *Optimization Letters* 11, pp. 17–30. DOI: 10.1007/s11590-016-1057-x.

Jackiewicz, Marcel, Adam Kasperski, and Paweł Zieliński (2024). "Recoverable Robust Shortest Path Problem Under Interval Budgeted Uncertainty Representations." In: *Networks* 85.1, pp. 127–141. DOI: 10.1002/net.22255.

Kempe, David, Jon Kleinberg, and Éva Tardos (2015). "Maximizing the Spread of Influence through a Social Network." In: *Theory of Computing* 11.4, pp. 105–147. DOI: 10.4086/toc.2015.v011a004.

Krause, Andreas and Carlos Guestrin (2007). "Near-optimal observation selection using submodular functions." In: *Proceedings of the 22nd national conference on Artificial intelligence-Volume 2*. AAAI'07, pp. 1650–1654.

Krause, Andreas, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta (2008). "Robust Submodular Observation Selection." In: *Journal of Machine Learning Research* 9.93, pp. 2761–2801.

Lachmann, Thomas, Stefan Lendl, and Gerhard J. Woeginger (2021). "A linear time algorithm for the robust recoverable selection problem." In: *Discrete Applied Mathematics* 303, pp. 94–107. DOI: 10.1016/j.dam.2020.08.012.

Liebchen, Christian, Marco Lübbecke, Rolf Möhring, and Sebastian Stiller (2009). "The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications." In: *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*. Ed. by Ravindra K. Ahuja, Rolf H. Möhring, and Christos D. Zaroliagis. Springer Berlin Heidelberg, pp. 1–27. DOI: 10.1007/978-3-642-05465-5_1.

Lin, Hui and Jeff Bilmes (2011). "A class of submodular functions for document summarization." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Springer, Cham, pp. 510–520.

Münch, Sabine, Stephen Raach, and Sven de Vries (2025). "Recoverable Robust Cardinality Constrained Maximization with Commitment of a Submodular Function." In: *Combinatorial Algorithms. IWOCA 2025*. Lecture Notes in Computer Science, vol 15885. Springer, pp. 376–390. DOI: 10.1007/978-3-031-98740-3_27.

Murota, Kazuo (2003). *Discrete Convex Analysis*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898718508.

– (2018). "A stronger multiple exchange property for $M^\natural$-concave functions." In: *Japan Journal of Industrial and Applied Mathematics* 35, pp. 411–421. DOI: 10.1007/s13160-017-0278-4.

Murota, Kazuo and Akiyoshi Shioura (2018). "Simpler exchange axioms for M-concave functions on generalized polymatroids." In: *Japan Journal of Industrial and Applied Mathematics* 35.1, pp. 235–259. DOI: 10.1007/s13160-017-0285-5.

Nemhauser, G. L., L. A. Wolsey, and M. L. Fisher (1978). "An analysis of approximations for maximizing submodular set functions-I." In: *Mathematical Programming* 14, pp. 265–294. DOI: 10.1007/BF01588971.

Orlin, James B., Andreas S. Schulz, and Rajan Udwani (2018). "Robust monotone submodular function maximization." In: *Mathematical Programming* 172, pp. 505–537. DOI: 10.1007/s10107-018-1320-2.

Paes Leme, Renato (2017). "Gross substitutability: An algorithmic survey." In: *Games and Economic Behavior* 106, pp. 294–316. DOI: 10.1016/j.geb.2017.10.016.
Tschiatschek, Sebastian, Rishabh Iyer, Haochen Wei, and Jeff Bilmes (2014). "Learning mixtures of submodular functions for image collection summarization." In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'14, pp. 1413–1421.
Wei, Kai, Rishabh Iyer, and Jeff Bilmes (2015). "Submodularity in data subset selection and active learning." In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, pp. 1954–1963.
Wolsey, Laurence A. (1982). "Maximising Real-Valued Submodular Functions: Primal and Dual Heuristics for Location Problems." In: *Mathematics of Operations Research* 7.3, pp. 410–425. DOI: 10.1287/moor.7.3.410.

A. Appendix: Omitted Proofs.

A.1. Proof of Lemma 3.

Proof. Since coverage functions are obviously normalized and monotone increasing, we only prove submodularity. Let E be a finite set and $E^1, \dots, E^n$ be a collection of subsets of E. Let $f\colon 2^{\{1,\dots,n\}} \to \mathbb{R}_{\geq 0}, X \mapsto |\bigcup_{i\in X} E^i|$ be a coverage function. Then, we have, for all $A \subseteq B \subseteq \{1, \dots, n\}$ and $j \in \{1, \dots, n\} \setminus B$, that

$$\begin{aligned} f(A \cup \{j\}) - f(A) &= |\bigcup_{i\in A\cup\{j\}} E^i| - |\bigcup_{i\in A} E^i| = |E^j \setminus \bigcup_{i\in A} E^i| \geq |E^j \setminus \bigcup_{i\in B} E^i| \\ &= |\bigcup_{i\in B\cup\{j\}} E^i| - |\bigcup_{i\in B} E^i| = f(B \cup \{j\}) - f(B). \end{aligned}$$

□

A.2. Proof of Lemma 8.

Lemma 8 was originally shown by Orlin et al. (2018) as part of the proof of Orlin et al. (2018, Theorem 12). Since Orlin et al.'s proof is very short and compact, we give a more detailed explanation here to improve understanding and accessibility.

Proof. Let $T \in \arg\max_{X\subseteq I\setminus D^0, |X|\leq k-w} f(X)$ and $H \in \arg\max_{X\subseteq I\setminus S^0, |X|\leq k-w} f(X)$. Define $T^0 := T \cap S^0$ and $T^C := T \setminus T^0$. Then, we have $T^0 \subseteq (I \setminus D^0) \cap S^0 = S^0 \setminus D^0$ and $T^C = T \setminus T^0 = T \setminus (T \cap S^0) = T \setminus S^0 \subseteq I \setminus S^0$. Furthermore, we have $f(H) \geq f(J)$ for all $J \subseteq I \setminus S^0$ with $|J| \leq k - w$, in particular for $J = T^C$. Hence, it follows from the monotonicity and submodularity of f that

$$f(S^0 \setminus D^0) + f(H) \geq f(T^0) + f(T^C) \geq f(T).$$

The claim follows by rearranging the above inequality. □

A.3. Proof of Lemma 9.

Lemma 9 was originally proved by Bogunovic et al. (2017). However, due to a small typesetting error in the original proof, we provide a corrected version here.

Proof. Let $i \in \{0, \dots, \lceil \log_2(w) \rceil\}$. Assume that $|G^{i,j(i)} \cap D| > 2^i$ for every $j(i) \in \left\{1, \dots, \left\lceil \frac{w}{2^i} \right\rceil\right\}$. Then,

$$|D| \geq |\bigcup_{j(i)=1}^{\lceil \frac{w}{2^i} \rceil} G^{i,j(i)} \cap D| > \left\lceil \frac{w}{2^i} \right\rceil 2^i \geq w,$$

which contradicts $|D| \leq w$. □

A.4. Proof of Lemma 10.

Proof. By submodularity of f, we have

$$f(D^1 | \bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D) \cup (S^1 \setminus D)) \leq f(D^1 | \bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D))$$
$$= f(D^1 | (G^{\lceil \log_2(w) \rceil, j(\lceil \log_2(w) \rceil)} \setminus D) \cup \bigcup_{i=0}^{\lceil \log_2(w) \rceil - 1} (G^{i,j(i)} \setminus D)).$$

Further, it follows from Bogunovic et al. (2017, Lemma D.6, Supplementary) and Bogunovic et al. (2017, Lemma D.7 with $\eta = 4\lceil \log_2(w) \rceil$, Supplementary) that f satisfies

$$f(\bigcup_{i=0}^{\lceil \log_2(w) \rceil} (G^{i,j(i)} \setminus D)) \geq \frac{1}{1+\alpha} f(G^{\lceil \log_2(w) \rceil, j(\lceil \log_2(w) \rceil)})$$

and

$$f(G^{\lceil \log_2(w) \rceil, j(\lceil \log_2(w) \rceil)} \cap D | \bigcup_{i=0}^{\lceil \log_2(w) \rceil - 1} (G^{i,j(i)} \setminus D)) \leq \alpha f(\bigcup_{i=0}^{\lceil \log_2(w) \rceil - 1} (G^{i,j(i)} \setminus D)),$$

where $\alpha \leq \frac{3}{4}$.

Therefore, and because S_1 is constructed in Line 6 of Algorithm 4 by applying Algorithm 1 to $(I, f, k - |S^0|)|_{I \setminus S^0}$ after determining the set $G^{\lceil \log_2(w) \rceil, \lceil \frac{w}{2^{\lceil \log_2(w) \rceil}} \rceil}$, we can apply Bogunovic et al. (2017, Lemma D.4, Supplementary) (with $X = \bigcup_{i=0}^{\lceil \log_2(w) \rceil - 1} (G^{i,j(i)} \setminus D)$, $Y = G^{\lceil \log_2(w) \rceil, j(\lceil \log_2(w) \rceil)}$, $Z = S^1$, $E_Y = D \cap Y$ and $E_Z = S^1 \cap D = D^1$) to obtain

$$f(D^1 | (G^{\lceil \log_2(w) \rceil, j(\lceil \log_2(w) \rceil)} \setminus D) \cup \bigcup_{i=0}^{\lceil \log_2(w) \rceil - 1} (G^{i,j(i)} \setminus D))$$

$$\leq \Big(\frac{|D^1|(1+\alpha)}{|G^{\lceil \log_2(w)\rceil, j(\lceil \log_2(w)\rceil)}|} + \alpha\Big) f\Big((G^{\lceil \log_2(w)\rceil, j(\lceil \log_2(w)\rceil)} \setminus D) \cup \bigcup_{i=0}^{\lceil \log_2(w)\rceil - 1} (G^{i,j(i)} \setminus D)\Big)$$

$$\leq \frac{19}{16} f\Big(\bigcup_{i=0}^{\lceil \log_2(w)\rceil} (G^{i,j(i)} \setminus D\Big),$$

where the last inequality follows by $\alpha \leq \frac{3}{4}$ and $\frac{|D^1|}{|G^{\lceil \log_2(w)\rceil, j(\lceil \log_2(w)\rceil)}|} \leq \frac{w}{2^{\lceil \log_2(w)\rceil} 4 \lceil \log_2(w)\rceil} \leq \frac{1}{4}$. In total, we have shown:

$$f\Big(D^1 \Big| \bigcup_{i=0}^{\lceil \log_2(w)\rceil} (G^{i,j(i)} \setminus D) \cup (S^1 \setminus D)\Big)$$

$$\leq f\Big(D^1 \Big| (G^{\lceil \log_2(w)\rceil, j(\lceil \log_2(w)\rceil)} \setminus D) \cup \bigcup_{i=0}^{\lceil \log_2(w)\rceil - 1} (G^{i,j(i)} \setminus D)\Big) \leq \frac{19}{16} f\Big(\bigcup_{i=0}^{\lceil \log_2(w)\rceil} (G^{i,j(i)} \setminus D\Big). \square$$

B. Appendix: Upper bound for the approximation factor of Algorithm 2.

Example 2. *For $n \in \mathbb{N}$, $n \geq 2$, let $S = \{s_1, \dots, s_{n+1}\}$, $A = \{a_1, \dots, a_{n+1}\}$, $A' = \{a'_1, \dots, a'_{n+1}\}$ and let $I = S \cup A \cup A'$. Further, let*

$$\sigma \colon 2^{A \cup A'} \to \mathbb{N},\ Y \mapsto |Y| - |\{j \in \{1, \dots, n+1\} \colon a_j \in Y,\ a'_j \in Y\}|$$

be the function that assigns to each set $Y \subseteq A \cup A'$ the cardinality of Y minus the cardinality of the set of all those indices that appear twice in Y.

Let $f^n \colon 2^I \to \mathbb{R}_{\geq 0}$ be defined by

$$f(\emptyset) \coloneqq 0,$$

$$f^n(s_i) \coloneqq \begin{cases} (n+1)\left(1 - \frac{1}{(1+\frac{1}{n})^n}\right) & \text{if } i = 1,\\ \frac{1}{(1+\frac{1}{n})^{i-2}} & \text{if } i \in \{2, \dots, n+1\}, \end{cases}$$

$$f^n(a) \coloneqq 2 - \frac{1}{(1+\frac{1}{n})^n} \text{ if } a \in A \cup A',$$

$$f^n(X) \coloneqq \sum_{x \in X} f^n(x) \text{ if } X \subseteq S,$$

$$f^n(X \cup Y) \coloneqq \sigma(Y) f^n(a_1) + f^n(X) - \sigma(Y)\Big(1 - \frac{1}{1+\frac{1}{n}}\Big) f^n(X) \text{ if } X \subseteq S \text{ and } Y \subseteq A \cup A'.$$

It is straightforward to prove that f^n is submodular. We first demonstrate for any $a \in A \cup A'$ and $j \in \{1, \dots, n+1\}$ that $f^n(a|\{s_1, \dots, s_{j-1}\}) = f^n(s_j|\{s_1, \dots, s_{j-1}\})$ by showing $f^n(\{a\} \cup \{s_1, \dots, s_{j-1}\}) = f^n(\{s_1, \dots, s_j\})$:

$$f^n(\{a\} \cup \{s_1, \dots, s_{j-1}\}) = f^n(a_1) + f^n(\{s_1, \dots, s_{j-1}\}) - \Big(1 - \frac{1}{1+\frac{1}{n}}\Big) f^n(\{s_1, \dots, s_{j-1}\})$$

$$
\begin{aligned}
&= 1 + (1 - \frac{1}{(1+\frac{1}{n})^n}) + \frac{1}{1+\frac{1}{n}} f^n(\{s_1, \ldots, s_{j-1}\}) \\
&= 1 + (1 - \frac{1}{(1+\frac{1}{n})^n}) + \frac{1}{1+\frac{1}{n}} \Bigg((n+1)(1 - \frac{1}{(1+\frac{1}{n})^n}) \\
&+ \sum_{i=2}^{j-1} \frac{1}{(1+\frac{1}{n})^{i-2}} \Bigg) \\
&= 1 + (n+1)(1 - \frac{1}{(1+\frac{1}{n})^n}) + \sum_{i=3}^{j} \frac{1}{(1+\frac{1}{n})^{i-2}} \\
&= f^n(s_1) + f^n(s_2) + \sum_{i=3}^{j} f^n(s_i) = f^n(\{s_1, \ldots, s_j\}).
\end{aligned}
$$

Furthermore, this implies for all $j \in \{1, \ldots, n+1\}$ *and* $x \in \{s_j, \ldots, s_{n+1}\}$ *that* $f^n(a|\{s_1, \ldots, s_{j-1}\}) = f^n(x|\{s_1, \ldots, s_{j-1}\})$*, since*

$$
f^n(a|\{s_1, \ldots, s_{j-1}\}) = f^n(s_j|\{s_1, \ldots, s_{j-1}\}) = f^n(s_j) > f^n(x) = f^n(x|\{s_1, \ldots, s_{j-1}\}).
$$

Thus, Algorithm 1 applied to the SMC*-instance* $(I, f^n, n+1)$ *returns the set* S*, when we assume that Algorithm 1 breaks ties in Line 3 by favoring any* $x \in S$ *over any* $a \in A \cup A'$*. Consequently, Algorithm 2 returns for the* RSMC*-instance* $(I, f^n, n+1, 1)$ *the initial solution* S *and the second-stage solution* $R = \emptyset$*. Then,* $(S, \{s_1\}, \emptyset)$ *is a solution tuple of* $(I, f^n, n+1, 1)$ *with* D *chosen adversarially to minimize the value of the corresponding ultimate solution.*

Furthermore, $\{a_1, \ldots, a_n\} \cup \{a'_{n+1}\}$ *is the ultimate solution corresponding to the optimal solution tuple* $(A, \{a_{n+1}\}, \{a'_{n+1}\})$ *and for the limit* $n \to \infty$*, we have*

$$
\lim_{n\to\infty} \frac{f^n(\{s_2, \ldots, s_{n+1}\})}{f^n(\{a_1, \ldots, a_n\} \cup \{a'_{n+1}\})} = \lim_{n\to\infty} \frac{\sum_{i=2}^{n+1} \frac{1}{(1+\frac{1}{n})^{i-2}}}{(n+1)(2 - \frac{1}{(1+\frac{1}{n})^n})} = \lim_{n\to\infty} \frac{(n+1)(1 - \frac{1}{(1+\frac{1}{n})^n})}{(n+1)(2 - \frac{1}{(1+\frac{1}{n})^n})}
$$
$$
= \frac{1 - e^{-1}}{2 - e^{-1}} = 0.387\ldots .
$$

Hence, the approximation factor of Algorithm 2 can not be better than $\frac{1-e^{-1}}{2-e^{-1}}$.

Notice that, even if we set – deviating from Algorithm 2 – the second-stage solution to $R = \{a_1\}$*, which is optimal for the initial solution* S *and deletion set* $\{s_1\}$*, we still cannot achieve a better approximation of the optimal ultimate solution, since*

$$
\begin{aligned}
\lim_{n\to\infty} \frac{f^n(\{a_1, s_2, \ldots, s_{n+1}\})}{f^n(\{a_1, \ldots, a_n\} \cup \{a'_{n+1}\})} &= \lim_{n\to\infty} \frac{2 - \frac{1}{(1+\frac{1}{n})^n} + \frac{1}{1+\frac{1}{n}} \sum_{i=2}^{n+1} \frac{1}{(1+\frac{1}{n})^{i-2}}}{(n+1)(2 - \frac{1}{(1+\frac{1}{n})^n})} \\
&= \lim_{n\to\infty} \frac{1 + (1 - \frac{1}{(1+\frac{1}{n})^n}) + \frac{n+1}{1+\frac{1}{n}}(1 - \frac{1}{(1+\frac{1}{n})^n})}{(n+1)(2 - \frac{1}{(1+\frac{1}{n})^n})}
\end{aligned}
$$

$$= \lim_{n\to\infty} \frac{1 + (1 - \frac{1}{(1+\frac{1}{n})^n}) + n(1 - \frac{1}{(1+\frac{1}{n})^n})}{(n+1)(2 - \frac{1}{(1+\frac{1}{n})^n})}$$

$$= \lim_{n\to\infty} \frac{1 - \frac{1}{(1+\frac{1}{n})^n}}{2 - \frac{1}{(1+\frac{1}{n})^n}} = \frac{1 - e^{-1}}{2 - e^{-1}} = 0.387\ldots .$$

Hence, no algorithm that determines an initial solution analogously to Algorithm 2 can achieve an approximation factor better than $\frac{1-e^{-1}}{2-e^{-1}}$.