# A Simple Deterministic Algorithm for Symmetric Submodular Maximization Subject to a Knapsack Constraint

Georgios Amanatidis[a,1,*], Georgios Birmpas[a,2], Evangelos Markakis[a]

[a] *Department of Informatics, Athens University of Economics and Business, Athens, Greece*

**Abstract**

We obtain a polynomial-time deterministic $\left(\frac{2e}{e-1}+\epsilon\right)$-approximation algorithm for maximizing symmetric submodular functions under a budget constraint. Although there exist randomized algorithms with better expected performance, our algorithm achieves the best known factor achieved by a deterministic algorithm, improving on the previously known factor of 6. Furthermore, it is simple, combining two elegant algorithms for related problems; the local search algorithm of Feige, Mirrokni and Vondrák [1] for unconstrained submodular maximization, and the greedy algorithm of Sviridenko [2] for non-decreasing submodular maximization subject to a knapsack constraint.

*Keywords:* Submodular maximization, symmetric submodular functions, knapsack constraints

## 1. Introduction

We study the problem of maximizing a *symmetric submodular* objective subject to a knapsack/budget constraint. A submodular function is called symmetric when the value of any set $S$ equals the value of its complement. Symmetric submodular functions form a prominent subclass of non-monotone submodular functions that includes *cut functions*, and has received considerable attention in the operations research literature, see, e.g., [3, 4].

For maximizing a *non-decreasing* submodular objective subject to a knapsack constraint there is a deterministic algorithm due to Sviridenko [2], achieving an approximation factor of $\frac{e}{e-1}$. This is the best possible factor in polynomial time, unless P = NP [5]. Sviridenko's algorithm is an adaptation of a greedy algorithm of Khuller et al. [6], which was designed for the special case of the budgeted maximum coverage problem.

Regarding maximization of non-monotone submodular functions subject to knapsack or other type of constraints, there is a vast literature, going back several decades, see, e.g., [7, 8]. More recently, Lee et al. [9] provided the first constant factor randomized algorithm for submodular maximization under $k$ matroid or $k$ knapsack constraints, with factors $k+2+\frac{1}{k}$ and 5 respectively. For $k$ knapsack constraints, Fadaei et al. [10] improved the heavy/light item approach of [9] and reduced the factor to 4. The problem was also studied by Gupta et al. [11] who proposed a (potentially randomized) algorithm, achieving a $(4+\alpha)$-approximation for a single knapsack constraint, where $\alpha$ is the approximation guarantee of any algorithm for the unconstrained submodular maximization problem, used here as a subroutine. In the case of symmetric submodular functions the algorithm of Gupta et al. [11], coupled, e.g., with the deterministic 2-approximation algorithm of Feige et al. [5] or that of Feldman [12], becomes a deterministic 6-approximation algorithm. In fact, before our work, this was the best known approximation factor for the problem achieved by a deterministic algorithm. Later on, Chekuri et al. [13] suggested a randomized 3.07-approximation algorithm improving the previously known results. Finally, Feldman et al. [14] and Kulik et al. [15] proposed their own randomized algorithms when there are knapsack constraints, achieving an $e$-approximation.[3]

Finally, combinations of local search and greedy algorithms have been used before in submodular function maximization. Fadaei et al. [10] used a variant of local search and the continuous greedy algorithm [16] for maximizing a non-monotone submodular function subject to packing polytope constraints. More recently, Sarpatwar et al. [17] combined Sviridenko's algorithm [2] with a variant of local search for maximizing a monotone submodular function subject to a knapsack and $k$ matroid constraints.

**Our contribution:** In this work we obtain a simple, deterministic $\left(\frac{2e}{e-1}+\epsilon\right)$-approximation algorithm for sym-

*Corresponding author

*Email addresses:* g.d.amanatidis@gmail.com (Georgios Amanatidis), gebirbas@gmail.com (Georgios Birmpas), markakis@aueb.gr (Evangelos Markakis)

[1] Current affiliations: *University of Essex*, Department of Mathematical Sciences, Colchester, United Kingdom, and *University of Amsterdam*, Institute for Logic, Language and Computation, Amsterdam, The Netherlands.

[2] Current affiliation: *Sapienza University of Rome*, Department of Computer, Control and Management Engineering, Rome, Italy.

[3] The algorithm of Kulik et al. [15] can be derandomized without any performance loss, but only by assuming an additional oracle for the extension by expectation, say $V$, of the objective function $v$. When only an oracle for $v$ is available, estimation of $V$ by sampling is required.

metric submodular maximization subject to a knapsack constraint. Note that $\frac{2e}{e-1} \approx 3.164$. This is the best known factor achieved by a deterministic algorithm assuming only a value oracle for the objective function. We design our algorithm by combining appropriate adaptations of two elegant well-known algorithms. The first one is the local search algorithm of Feige et al. [1] that was designed for unconstrained (symmetric) submodular maximization. The second one is Sviridenko's algorithm [2] for non-decreasing submodular maximization subject to a knapsack constraint. From a technical point of view, our use of local search in order to identify regions of the function's domain, where results for non-decreasing submodular functions can be directly applied is novel.

To see why local search comes handy for a symmetric submodular function $v(\cdot)$, suppose we could produce two local optima, namely a set $S$ and its complement. It is not hard to prove that the restriction of $v(\cdot)$ to each local optimum is a *non-decreasing* submodular function. This allows us to utilize Sviridenko's algorithm [2] on the two subsets and then show that one of the two solutions attains a good approximation. The running time is still an issue under this approach, however, since finding exact local optima is not guaranteed to run in polynomial time [18]. What comes to rescue is that even by finding *approximate* local optima, the objective function, restricted within each of them, remains *almost non-decreasing* in a certain sense. This way we are still able to appropriately adjust Sviridenko's algorithm [2] with only a small loss. This *"robustness under small deviations from monotonicity"* approach that allows the use of known results for monotone objectives was introduced by the authors in [19]. There the main application was the design of truthful, budget-feasible mechanisms.

## 2. Notation and Preliminaries

We use $A = [n] = \{1, 2, ..., n\}$ to denote a set of $n$ items. Each item $i$ is associated with an integral cost $c_i$. There is also a valuation function $v : 2^A \to \mathbb{Q}_{\geq 0}$ and an integral budget $B > 0$. For $S \subseteq A$, $v(S)$ is the value derived if the set $S$ is selected (for singletons, we will often write $v(i)$ instead of $v(\{i\})$). Therefore, the goal is to select a set $S$ that maximizes $v(S)$ subject to the constraint $\sum_{i \in S} c_i \leq B$. We assume oracle access to $v$ via value queries, i.e., we assume the existence of a polynomial time value oracle that returns $v(S)$ when given as input a set $S$.

We focus on a natural subclass of submodular valuation functions that includes cut functions, namely non-negative symmetric submodular functions. Throughout this work we make the natural assumption that $v(\emptyset) = 0$.

**Definition 2.1.** *A function $v$, defined on $2^A$ for some set $A$, is*

- submodular, *if $v(S \cup \{i\}) - v(S) \geq v(T \cup \{i\}) - v(T)$ for any $S \subseteq T \subseteq A$, and $i \notin T$,*
- non-decreasing, *if $v(S) \leq v(T)$ for any $S \subseteq T \subseteq A$,*
- symmetric *if $v(S) = v(A \setminus S)$ for any $S \subseteq A$.*

It is easy to see that $v$ cannot be both symmetric and non-decreasing unless it is a constant function. In fact, if this is the case and $v(\emptyset) = 0$, then $v(S) = 0$, for all $S \subseteq A$. We also state an alternative definition of a submodular function, which will be useful later.

**Theorem 2.2** (Nemhauser et al. [7])**.** *A set function $v$ is submodular if and only if for all $S, T \subseteq A$ we have*

$$v(T) \leq v(S) + \sum_{i \in T \setminus S} (v(S \cup \{i\}) - v(S)) - \sum_{i \in S \setminus T} (v(S \cup T) - v(S \cup T \setminus \{i\})) .$$

We often need to argue about optimal solutions of subinstances, from an instance we begin with. Given a cost vector $\mathbf{c}$, and a subset $X \subseteq A$, we denote by $\mathbf{c}_X$ the projection of $\mathbf{c}$ on $X$. We also let $\text{OPT}(X, v, \mathbf{c}_X, B)$ be the value of an optimal solution to the restriction of this instance on $X$, i.e., $\text{OPT}(X, v, \mathbf{c}_X, B) = \max_{S:\, S \subseteq X,\, \mathbf{c}(S) \leq B} v(S)$. Similarly, $\text{OPT}(X, v, \mathbf{c}_X, \infty)$ denotes the value of an optimal solution to the unconstrained version of the problem restricted on $X$. For the sake of readability, we usually drop the valuation function and the cost vector, when they are clear from context, and write $\text{OPT}(X, B)$ or $\text{OPT}(X, \infty)$.

Finally, we make one further assumption: we assume there is at most one item whose cost exceeds the budget. As shown in Lemma 3.1, this is without loss of generality.

**Local Optima and Local Search.** Given $v : 2^A \to \mathbb{Q}$, a set $S \subseteq A$ is called a $(1+\epsilon)$-*approximate local optimum* of $v$, if $(1+\epsilon)v(S) \geq v(S \setminus \{i\})$ and $(1+\epsilon)v(S) \geq v(S \cup \{i\})$ for every $i \in A$. When $\epsilon = 0$, $S$ is called an *exact local optimum* of $v$. Note that if $v$ is symmetric submodular, then $S$ is a $(1+\epsilon)$-approximate local optimum if and only if $A \setminus S$ is a $(1+\epsilon)$-approximate local optimum.

Approximate local optima produce good approximations in unconstrained maximization of general submodular functions [1]. However, here they are of interest for a quite different reason that becomes apparent in Lemma 4.1. We can efficiently find approximate local optima using the local search algorithm ApproxLS of [1]. Note that this is an algorithm for the unconstrained version of the problem, when there is no budget constraint.

If we care to find an exact local optimum, we can simply set $\epsilon = 0$. In this case, however, we cannot argue about the running time of the algorithm in general.

**Lemma 2.3** (inferred from [1])**.** *Given a submodular function $v : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ and a value oracle for $v$,* ApproxLS$(A, v, \epsilon)$ *outputs a $\left(1 + \frac{\epsilon}{n^2}\right)$-approximate local optimum using $O\left(\frac{1}{\epsilon} n^3 \log n\right)$ calls to the oracle.*

## 3. Instances with Costs Exceeding the Budget

Consider an instance with a symmetric submodular objective function, where there exist items $i$ with $c_i > B$;

ApproxLS$(A, v, \epsilon)$ [1]

$S = \{i^*\}$, where $i^* \in \arg\max_{i \in A} v(i)$
**while** *there exists some a such that* $\max\{v(S \cup \{a\}), v(S \setminus \{a\})\} > (1 + \epsilon/n^2)v(S)$ **do**
  **if** $v(S \cup \{a\}) > (1 + \epsilon/n^2)v(S)$ **then**
    $S = S \cup \{a\}$
  **else**
    $S = S \setminus \{a\}$
**return** $S$

we call such items *expensive*. The presence of expensive items can create infeasible solutions of very high value and make an analog of Lemma 4.3 (and thus of Lemma 4.1) impossible to prove. At first glance, it may seem reasonable to just discard expensive items, since they are not included in any feasible solution anyway. However, simply discarding them could destroy the symmetry of the function, since it may no longer be guaranteed that $v(S) = v(A' \setminus S)$, where $A'$ is the new shrunk set of items.

Let $\mathcal{I}$ denote the set of all instances of the problem with a symmetric submodular objective, and let $\mathcal{J} \subseteq \mathcal{I}$ denote the set of all instances with at most one expensive item, i.e., $\mathcal{J}$ contains the instances where at most one item has cost more than $B$. Given $X \subseteq A$, we let $\mathbf{c}(X) = \sum_{i \in X} c_i$. The next lemma, together with its corollary, show that, when dealing with symmetric submodular functions, we may only consider instances in $\mathcal{J}$ without any loss of generality.

**Lemma 3.1.** *Given an instance $I = (A, v, \mathbf{c}, B) \in \mathcal{I}$, we can efficiently construct an instance $J = (A', v', \mathbf{c}', B) \in \mathcal{J}$ such that:*

(i) *Every feasible solution of $I$ is a feasible solution of $J$ and vice versa.*
(ii) *If $X$ is a feasible solution of $I$, then $v(X) = v'(X)$ and $\mathbf{c}(X) = \mathbf{c}'(X)$. In particular,* $\textsc{opt}(J) = \textsc{opt}(I)$.
(iii) *The function $v'$ (defined on the set $A'$) is symmetric submodular.*

*Proof.* Let $E = \{i \in A \mid c_i > B\}$ be the set of expensive items and define $A' = (A \setminus E) \cup \{i_E\}$, where $i_E$ is a new item replacing the whole set $E$. For $i \in A \setminus E$ we define $c'_i = c_i$, while $c'_{i_E} = B + 1$. Finally, $v'$ is defined on subsets of $A'$ as follows

$$v'(T) = \begin{cases} v(T), & \text{if } T \subseteq A \setminus E, \\ v((T \setminus \{i_E\}) \cup E), & \text{otherwise}. \end{cases}$$

Now suppose $X$ is a budget-feasible solution of $I$. Then $\mathbf{c}(X) \leq B$ and thus $X \subseteq A \setminus E$. But then, by the definition of $\mathbf{c}'$, $\mathbf{c}'(X) = \mathbf{c}(X) \leq B$ as well, and therefore $X$ is also a budget-feasible solution of $J$. Moreover, $v'(X) = v(X)$ by the definition of $v'$. We conclude that $\textsc{opt}(I) \leq \textsc{opt}(J)$.

The proof that every feasible solution of $J$ is a feasible solution of $I$ is almost identical. This implies $\textsc{opt}(J) \leq \textsc{opt}(I)$, and therefore $\textsc{opt}(J) = \textsc{opt}(I)$.

To see that $v'$ is symmetric, consider a set $S \subseteq A'$. Without loss of generality, we may assume that $i_E \in S$ (or else we just exchange the roles of $S$ and $A' \setminus S$). By the definition of $v'$ and the fact that $v$ is symmetric, we have

$$v'(S) = v((S \setminus \{i_E\}) \cup E) = v(A \setminus (S \cup E)) = v'(A' \setminus S)\,.$$

Finally, we show the submodularity of $v'$. Let $S \subseteq T \subseteq A'$. We are going to consider a few cases with respect to $i_E$. If $i_E \in S \cap T$, then, by the definition of $v'$ and the submodularity of $v$, we have that for any $i \notin T$,

$$\begin{aligned} v'(S \cup \{i\}) - v'(S) &= v(S \cup \{i\}) - v(S) \\ &\geq v(T \cup \{i\}) - v(T) \\ &= v'(T \cup \{i\}) - v'(T)\,. \end{aligned}$$

Similarly, if $i_E \in T$ but $i_E \notin S$, then for any $i \notin T$,

$$\begin{aligned} v'(S \cup \{i\}) - v'(S) &= v(S \cup \{i\}) - v(S) \\ &\geq v(((T \setminus \{i_E\}) \cup E) \cup \{i\}) \\ &\quad - v((T \setminus \{i_E\}) \cup E) \\ &= v'(T \cup \{i\}) - v'(T)\,. \end{aligned}$$

When $i_E \notin T$, we consider two subcases for the different $i \notin T$. First, for any $i \notin T, i \neq i_E$ exactly the same equalities and inequalities as in the first case hold. Second, for $i = i_E$ assume that $E = \{i_1, \ldots, i_{|E|}\}$ and let $E_0 = \emptyset$ and $E_k = \{i_1, \ldots, i_k\}$ for $k \in [|E|]$. We have,

$$\begin{aligned} v'(S \cup \{i_E\}) - v'(S) &= v(S \cup E) - v(S) \\ &= \sum_{k=1}^{|E|} v(S \cup E_k) - v(S \cup E_{k-1}) \\ &\geq \sum_{k=1}^{|E|} v(T \cup E_k) - v(T \cup E_{k-1}) \\ &= v(T \cup E) - v(T) \\ &= v'(T \cup \{i_E\}) - v'(T)\,. \end{aligned}$$

To see why the inequality holds, note that for each $k \in [|E|]$, we have $S \cup E_{k-1} \subseteq T \cup E_{k-1} \subseteq A$ and $i_k \notin T \cup E_{k-1}$. Thus, by the submodularity of $v$, we get $v(S \cup E_k) - v(S \cup E_{k-1}) \geq v(T \cup E_k) - v(T \cup E_{k-1})$. □

Now, it is not hard to see that we can turn any (approximate) algorithmic result on $\mathcal{J}$ to the same algorithmic result on $\mathcal{I}$. This is summarized in the following corollary.

**Corollary 3.2.** *Given a polynomial-time algorithm* ALG′ *that achieves a $\rho$-approximation on instances in $\mathcal{J}$, we can efficiently construct a polynomial-time $\rho$-approximation algorithm* ALG *that works for all instances in $\mathcal{I}$.*

*Proof.* The description of ALG is quite straightforward. Given an instance $I = (A, v, \mathbf{c}, B) \in \mathcal{I}$, ALG first constructs instance $J = (A', v', \mathbf{c}', B) \in \mathcal{J}$, as described in the proof of Lemma 3.1. Then ALG runs ALG′ with input $J$ and returns its output. Clearly, if ALG′ runs in polynomial

time, so does ALG. If $X = \text{ALG}'(J) = \text{ALG}(I)$, then $X$ is feasible with respect to $J$ and $\text{OPT}(J) \leq \rho \cdot v'(X)$. By Lemma 3.1 we get that $X$ is feasible with respect to $I$ and $\text{OPT}(I) = \text{OPT}(J) \leq \rho \cdot v'(X) = \rho \cdot v(X)$. This establishes the approximation ratio of ALG. □

## 4. A Simple Algorithm

The main result of this section is a polynomial-time, deterministic $\left(\frac{2e}{e-1} + \epsilon\right)$-approximation algorithm for symmetric submodular functions. Throughout the section we assume there is at most one item with cost exceeding $B$.

Since our function is not monotone, we cannot directly apply the result of [2], which gives an optimal greedy algorithm for non-decreasing submodular maximization subject to a knapsack constraint. Instead, our main idea is to combine appropriately the result of [2] with the local search used for unconstrained symmetric submodular maximization [1]. At a high level, what happens is that local search produces an approximate solution $S$ for the unconstrained problem, and while this does not look related to our goal at first sight, $v$ is "close to being non-decreasing" on both $S$ and $A \setminus S$. This becomes precise in Lemma 4.1 below, but the point is that running a modification of the greedy algorithm of [2], on both $S$ and $A \setminus S$ will now produce at least one good enough solution.

LS-GREEDY$(A, v, \mathbf{c}, B, \epsilon)$

1 $S = \text{APPROXLS}(A, v, \epsilon/4)$
2 $T_1 = \text{VARGREEDY}(S, v, \mathbf{c}_S, B)$
3 $T_2 = \text{VARGREEDY}(A \setminus S, v, \mathbf{c}_{A \setminus S}, B)$
4 Let $T$ be the best solution among $T_1$ and $T_2$
5 **return** $T$

The first component of our algorithm is the local search algorithm of [1]. By Lemma 2.3 and the fact that $v$ is symmetric, both $S$ and $A \setminus S$ are $\left(1 + \frac{\epsilon}{4n^2}\right)$-approximate local optima. We can now quantify the crucial observation that $v$ is close to being non-decreasing within the approximate local optima $S$ and $A \setminus S$. Actually, we only need this property on the local optimum that contains the best feasible solution. More precisely, the next lemma deals with the restriction of $v$ to an approximate local optimum $X$ that contains a feasible solution of value comparable to that of the globally optimal solution. (Note that any approximate local optimum or its complement satisfy the latter property.) Then any negative marginal value—of any item and with respect to any subset of $X$—is *tiny* compared to the value of an optimal solution within $X$.

**Lemma 4.1.** *Let $S$ be a $\left(1 + \frac{\epsilon}{4n^2}\right)$-approximate local optimum and consider*

$$X \in \operatorname*{arg\,max}_{Z \in \{S, A \setminus S\}} \text{OPT}(Z, B)\,.$$

*Then, for every $T \subsetneq X$ and every $i \in X \setminus T$, we have $v(T \cup \{i\}) - v(T) \geq -\frac{\epsilon}{n} \text{OPT}(X, B)$.*

Before proving Lemma 4.1, we begin with a simple fact and a useful lemma. We note that Fact 4.2 and Lemma 4.3 below require only subadditivity. Submodularity is used later, within the proof of Lemma 4.1.

**Fact 4.2.** *For any $S \subseteq A$, $\max\{\text{OPT}(S, B), \text{OPT}(A \setminus S, B)\} \geq 0.5\,\text{OPT}(A, B)$ since $\text{OPT}(S, B) + \text{OPT}(A \setminus S, B) \geq \text{OPT}(A, B)$ by subadditivity.*

**Lemma 4.3.** *For any $S \subseteq A$, $\text{OPT}(S, \infty) \leq 2n \cdot \text{OPT}(A, B)$.*

*Proof.* Let $S^* \subseteq S$ be such that $v(S^*) = \text{OPT}(S, \infty)$. By subadditivity we have $\text{OPT}(S, \infty) = v(S^*) \leq \sum_{i \in S^*} v(i)$. Consider three cases with respect to $\{i \in A \mid c_i > B\}$.

If $\{i \in A \mid c_i > B\} = \emptyset$, then by the fact that every singleton is a feasible solution we have $\sum_{i \in S^*} v(i) \leq n \cdot \max_{i \in A} v(i) \leq n \cdot \text{OPT}(A, B)$.

If $\{i \in A \mid c_i > B\} = \{x\} \not\subseteq S^*$, then every singleton in $A \setminus \{x\}$ is a feasible solution, and like before we have $\sum_{i \in S^*} v(i) \leq (n-1) \cdot \max_{i \in A \setminus \{x\}} v(i) \leq (n-1) \cdot \text{OPT}(A, B)$.

If $\{i \in A \mid c_i > B\} = \{x\} \subseteq S^*$, then we need to bound $v(x)$. Since $v$ is symmetric we have $v(x) = v(A \setminus \{x\}) \leq \sum_{i \in A \setminus \{x\}} v(i) \leq (n-1) \cdot \max_{i \in A \setminus \{x\}} v(i)$. Therefore, by using again that every singleton in $A \setminus \{x\}$ is a feasible solution, we have $\sum_{i \in S^*} v(i) \leq v(x) + (n-1) \cdot \max_{i \in A \setminus \{x\}} v(i) \leq (2n-2) \cdot \max_{i \in A \setminus \{x\}} v(i) \leq 2n \cdot \text{OPT}(A, B)$. □

*Proof of Lemma 4.1.* By Fact 4.2 we have $\text{OPT}(X, B) \geq 0.5\,\text{OPT}(A, B)$. Let $T \subseteq X \setminus \{i\}$ for some $i \in X$. By submodularity we have $v(T \cup \{i\}) - v(T) \geq v(X) - v(X \setminus \{i\})$. Since $S$ is a $\left(1 + \frac{\epsilon}{4n^2}\right)$-approximate local optimum and $v$ is symmetric, $X$ is also a $\left(1 + \frac{\epsilon}{4n^2}\right)$-approximate local optimum. As a result, $v(X \setminus \{i\}) \leq \left(1 + \frac{\epsilon}{4n^2}\right) v(X)$ and thus $v(X) - v(X \setminus \{i\}) \geq -\frac{\epsilon}{4n^2} v(X) \geq -\frac{\epsilon}{4n^2}\,\text{OPT}(X, \infty) \geq -\frac{\epsilon}{2n}\,\text{OPT}(A, B) \geq -\frac{\epsilon}{n}\,\text{OPT}(X, B)$, where the third inequality follows from Lemma 4.3. □

The second component of LS-GREEDY is an appropriate modification of the greedy algorithm of [2]. It first enumerates all solutions of size at most 3. Then, starting from each feasible 3-set, it builds a greedy solution, and it outputs the best among these $\Theta(n^3)$ solutions. Here this idea is adjusted for non-monotone submodular functions.

By Fact 4.2, at least one of $S$ and $A \setminus S$ contains a feasible solution of value at least $0.5\,\text{OPT}(A, B)$. Lemma 4.1 guarantees that within this set, $v$ is very close to a non-decreasing submodular function. This is sufficient for VARGREEDY to perform almost as well as if $v$ was non-decreasing.

**Theorem 4.4.** *For any $\epsilon > 0$, algorithm* LS-GREEDY *achieves a $\left(\frac{2e}{e-1} + \epsilon\right)$-approximation.*

*Proof.* Recall that VARGREEDY runs on both $S$ and $A \setminus S$ and LS-GREEDY returns the best solution of these two. We may assume, without loss of generality that $\text{OPT}(S, B) = \max\{\text{OPT}(S, B), \text{OPT}(A \setminus S, B)\}$ (the case for

VarGreedy$(A, v, \mathbf{c}, B)$

1 Let $S_1$ be the best feasible solution of cardinality at most 3 (by enumerating them all)
2 $S_2 = \emptyset$
3 **for** *every* $U \subseteq A$ *with* $|U| = 3$ *and* $\sum_{i \in U} c_i \leq B$ **do**
4 $\quad S^0 = U,\ t = 1,\ A^0 = A \setminus U$
5 $\quad$ **while** $A^{t-1} \neq \emptyset$ **do**
6 $\quad\quad$ Find $m_t = \max_{i \in A^{t-1}} \dfrac{v(S^{t-1} \cup \{i\}) - v(S^{t-1})}{c_i}$
7 $\quad\quad$ Let $i_t$ be an element of $A^{t-1}$ that attains $m_t$
8 $\quad\quad$ **if** $m_t \geq 0$ *and* $\sum_{i \in S^{t-1} \cup \{i_t\}} c_i \leq B$ **then**
9 $\quad\quad\quad$ $S^t = S^{t-1} \cup \{i_t\}$
10 $\quad\quad$ **else**
11 $\quad\quad\quad$ $S^t = S^{t-1}$
12 $\quad\quad$ $A^t = A^{t-1} \setminus \{i_t\}$
13 $\quad\quad$ $t = t + 1$
14 $\quad$ **if** $v(S^{t-1}) > v(S_2)$ **then**
15 $\quad\quad$ $S_2 = S^{t-1}$
16 **return** $S \in \arg\max_{X \in \{S_1, S_2\}} v(X)$

$A \setminus S$ being symmetric). By Fact 4.2 we have $\text{OPT}(S, B) \geq 0.5\,\text{OPT}(A, B)$. So, it suffices to show that running VarGreedy on $S$ outputs a set of value at least $(1 - 1/e - \epsilon)\,\text{OPT}(S, B)$.

In what follows we analyze the approximation ratio of VarGreedy$(S, v, \mathbf{c}, B)$ with respect to $\text{OPT}(S, B)$. For this, we follow closely the proof of the main result in [2].

If there is an optimal solution for our problem restricted on $S$, of cardinality one, two or three, then the set $S_1$ produced by VarGreedy will be such a solution. Hence, assume that the cardinality of any optimal solution is at least four and let $S^*$ be such a solution. If necessary, reorder the elements of $S^* = \{j_1, \ldots, j_{|S^*|}\}$ so that $j_1 = \arg\max_\ell v(\{j_\ell\})$, and $j_{k+1} = \arg\max_{\ell > k}[v(\{j_1, \ldots, j_k, j_\ell\}) - v(\{j_1, \ldots, j_k\})]$. Let $Y = \{j_1, j_2, j_3\}$.

For notational convenience, we will use the function $g(\cdot) = v(\cdot) - v(Y)$. It is straightforward that $g(\cdot)$ is submodular. Moreover, the following fact follows from [2].

**Fact 4.5.** *$g(X \cup \{i\}) - g(X) \leq \frac{1}{3} v(Y)$, for any $Y \subseteq X \subseteq S$ and $i \in S^* \setminus X$.*

Consider the execution of lines 3–13 of VarGreedy with initial set $U = Y$. Let $t^* + 1$ be the first time when an element $i_{t^*+1} \in S^*$ is not added to $S^{t^*}$. In fact, we assume that $t^* + 1$ is the first time when $S^t = S^{t-1}$. To see that this is without loss of generality, if there is some time $\tau < t^* + 1$ such that $i_\tau$ is not added to $S^{\tau-1}$, then—by the definition of $t^* + 1$—it must be the case that $i_\tau \notin S^*$. But then, we may consider the instance $(S \setminus \{i_\tau\}, v, \mathbf{c}_{S \setminus \{i_\tau\}}, B)$ instead. We have $v(S^*) = \text{OPT}(S \setminus \{i_\tau\}, B) = \text{OPT}(S, B)$ and the greedy solution constructed in the iteration where $S^0 = Y$ is exactly the same as before. We are going to distinguish two cases.

**Case 1.** For all $t \in [t^*], m_t \geq 0$, but $m_{t^*+1} < 0$. Using Theorem 2.2 for $S^*$ and $S^{t^*}$ we have

$$\begin{aligned}
g(S^*) &\leq g(S^{t^*}) + \sum_{i \in S^* \setminus S^{t^*}} (g(S^{t^*} \cup \{i\}) - g(S^{t^*})) \\
&\quad - \sum_{i \in S^{t^*} \setminus S^*} (g(S^{t^*} \cup S^*) - g(S^{t^*} \cup S^* \setminus \{i\})) \\
&= g(S^{t^*}) + \sum_{i \in S^* \setminus S^{t^*}} (v(S^{t^*} \cup \{i\}) - v(S^{t^*})) \\
&\quad - \sum_{i \in S^{t^*} \setminus S^*} (v(S^{t^*} \cup S^*) - v(S^{t^*} \cup S^* \setminus \{i\})) \\
&\leq g(S^{t^*}) + \sum_{i \in S^* \setminus S^{t^*}} c_i m_{t^*+1} \\
&\quad - |S^{t^*} \setminus S^*| \left(-\frac{\epsilon}{n}\,\text{OPT}(S, B)\right) \\
&\leq g(S^{t^*}) + \epsilon\,\text{OPT}(S, B)\,,
\end{aligned}$$

Here, the second to last inequality holds by Lemma 4.1 and by the assumptions we have made. That is, for every $i \in S^* \setminus S^{t^*}$, we have that $i \in A^{t^*}$, since we assumed that $t^* + 1$ is the first time when $S^t = S^{t-1}$. Hence, up until time $t^*$, $A^{t^*}$ contains all the items apart from $S^{t^*}$. This implies that for every $i \in S^* \setminus S^{t^*}$, we have that $v(S^{t^*} \cup \{i\}) - v(S^{t^*}) \leq c_i m_{t^*+1}$, by the definition of $m_{t^*+1}$.

Therefore, we may conclude that

$$\begin{aligned}
v(S^{t^*}) &= v(Y) + g(S^{t^*}) \\
&\geq v(Y) + g(S^*) - \epsilon\,\text{OPT}(S, B) \\
&= (1 - \epsilon)\,\text{OPT}(S, B)\,.
\end{aligned}$$

**Case 2.** For all $t \in [t^*+1], m_t \geq 0$, but $\sum_{i \in S^{t^*} \cup \{i_{t^*+1}\}} c_i > B$ while $\sum_{i \in S^{t^*}} c_i \leq B$. Using Theorem 2.2 for $S^*$ and each of $S^t$, $t \in [t^*] \cup \{0\}$, as well as Lemma 4.1, we have

$$\begin{aligned}
g(S^*) &\leq g(S^t) + \sum_{i \in S^* \setminus S^t} (g(S^t \cup \{i\}) - g(S^t)) \\
&\quad - \sum_{i \in S^t \setminus S^*} (g(S^t \cup S^*) - g(S^t \cup S^* \setminus \{i\})) \\
&= g(S^t) + \sum_{i \in S^* \setminus S^t} (v(S^t \cup \{i\}) - v(S^t)) \\
&\quad - \sum_{i \in S^t \setminus S^*} (v(S^t \cup S^*) - v(S^t \cup S^* \setminus \{i\})) \\
&\leq g(S^t) + \sum_{i \in S^* \setminus S^t} (v(S^t \cup \{i\}) - v(S^t)) \\
&\quad - |S^t \setminus S^*| \left(-\frac{\epsilon}{n}\,\text{OPT}(S, B)\right) \\
&\leq g(S^t) + \sum_{i \in S^* \setminus S^t} (v(S^t \cup \{i\}) - v(S^t)) \\
&\quad + \epsilon\,\text{OPT}(S, B)\,,
\end{aligned}$$

and therefore

$$g(S^*) - \epsilon\,\text{OPT}(S, B) \leq g(S^t) + \sum_{i \in S^* \setminus S^t} (v(S^t \cup \{i\}) - v(S^t))$$

$$\leq g(S^t) + \sum_{i \in S^* \setminus S^t} c_i m_{t+1}$$
$$\leq g(S^t) + \left(B - \sum_{i \in Y} c_i\right) m_{t+1}\,,$$

for all $t \in [t^*] \cup \{0\}$.

For the last part of the proof we need the following inequality of Wolsey [8].

**Theorem 4.6** (Wolsey [8])**.** *Let $k$ and $s$ be arbitrary positive integers, and $\rho_1, \ldots, \rho_k$ be arbitrary reals with $z_1 = \sum_{i=1}^{k} \rho_i$ and $z_2 = \min_{x \in [k]} \left(\sum_{i=1}^{x-1} \rho_i + s\rho_x\right) > 0$. Then $z_1/z_2 \geq 1 - (1 - 1/s)^k \geq 1 - e^{-\frac{k}{s}}$.*

For any $t \geq 1$, define $B_t = \sum_{\tau=1}^{t} c_{i_\tau}$, and let $B_0 = 0$, $k = B_{t^*+1}$ and $s = B - \sum_{i \in Y} c_i$. Also define $\rho_1, \ldots, \rho_k$ so that for $B_t < i \leq B_{t+1}$, $\rho_i = m_{t+1}$. It is easy to see that

$$g(S^{t^*} \cup \{i_{t^*+1}\}) = \sum_{\tau=1}^{t^*+1} c_{i_\tau} m_{i_\tau} = \sum_{i=1}^{k} \rho_i\,,$$

and similarly, for each of $S^t$, $t \in [t^*] \cup \{0\}$,

$$g(S^t) = \sum_{\tau=1}^{t} c_{i_\tau} m_{i_\tau} = \sum_{i=1}^{B_t} \rho_i\,.$$

As noted in [2], since the $\rho_i$s are nonnegative we have

$$\min_{x \in [k]} \left(\sum_{i=1}^{x-1} \rho_i + s\rho_x\right) = \min_{t \in [t^*] \cup \{0\}} \left(\sum_{i=1}^{B_t} \rho_i + s\rho_{B_t+1}\right),$$

and therefore

$$g(S^*) - \epsilon \operatorname{OPT}(S, B) \leq \min_{x \in [k]} \left(\sum_{i=1}^{x-1} \rho_i + s\rho_x\right).$$

So, as a direct application of Theorem 4.6 we have

$$\frac{g(S^{t^*} \cup \{i_{t^*+1}\})}{g(S^*) - \epsilon \operatorname{OPT}(S, B)} \geq 1 - e^{-\frac{k}{s}} > 1 - e^{-1}\,.$$

Finally, using the above inequality and Fact 4.5, we get

$$\begin{aligned}
v(S^{t^*}) &= v(Y) + g(S^{t^*}) \\
&= v(Y) + g(S^{t^*} \cup \{i_{t^*+1}\}) \\
&\quad - \big(g(S^{t^*} \cup \{i_{t^*+1}\}) - g(S^{t^*})\big) \\
&\geq v(Y) + (1 - e^{-1}) g(S^*) - (1 - e^{-1}) \epsilon \operatorname{OPT}(S, B) \\
&\quad - \big(v(S^{t^*} \cup \{i_{t^*+1}\}) - v(S^{t^*})\big) \\
&\geq v(Y) + (1 - e^{-1}) g(S^*) - \epsilon \operatorname{OPT}(S, B) - \frac{1}{3} v(Y) \\
&\geq (1 - e^{-1} - \epsilon) \operatorname{OPT}(S, B)\,.
\end{aligned}$$

Since in both cases the final output $T^*$ of the algorithm has value at least $v(S^{t^*})$, this implies that

$$v(T^*) \geq (1 - e^{-1} - \epsilon) \operatorname{OPT}(S, B)$$
$$\geq \frac{1 - e^{-1} - \epsilon}{2} \operatorname{OPT}(A, B)$$
$$> \left(\frac{e - 1}{2e} - \epsilon\right) \operatorname{OPT}(A, B)\,,$$

thus concluding the analysis of the performance of the algorithm. □

Note that the number of oracle value queries of VARGREEDY, i.e., $O(n^5)$, dominates the number of value queries of LS-GREEDY for constant $\epsilon$.

## Acknowledgments

This work was partially supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support faculty members and researchers and the procurement of high-cost research equipment grant" (Project Number: 3512).

## References

[1] U. Feige, V. S. Mirrokni, J. Vondrák, Maximizing non-monotone submodular functions, SIAM J. Comput. 40 (4) (2011) 1133–1153.

[2] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, Oper. Res. Lett. 32 (1) (2004) 41–43.

[3] S. Fujishige, Canonical decompositions of symmetric submodular systems, Discrete Applied Mathematics 5 (1983) 175–190.

[4] M. Queyranne, Minimizing symmetric submodular functions, Mathematical Programming 82 (1-2) (1998) 3–12.

[5] U. Feige, A threshold of ln $n$ for approximating set cover, J. ACM 45 (4) (1998) 634–652.

[6] S. Khuller, A. Moss, J. Naor, The budgeted maximum coverage problem, Inf. Process. Lett. 70 (1) (1999) 39–45.

[7] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An analysis of approximations for maximizing submodular set functions - I, Math. Program. 14 (1) (1978) 265–294.

[8] L. A. Wolsey, Maximising real-valued submodular functions: Primal and dual heuristics for location problems, Math. Oper. Res. 7 (3) (1982) 410–425.

[9] J. Lee, V. S. Mirrokni, V. Nagarajan, M. Sviridenko, Maximizing nonmonotone submodular functions under matroid or knapsack constraints, SIAM J. Discrete Math. 23 (4) (2010) 2053–2078.

[10] S. Fadaei, M. Fazli, M. Safari, Maximizing non-monotone submodular set functions subject to different constraints: Combined algorithms, Oper. Res. Lett. 39 (6) (2011) 447–451.

[11] A. Gupta, A. Roth, G. Schoenebeck, K. Talwar, Constrained non-monotone submodular maximization: Offline and secretary algorithms, in: Proceedings of the 6th International Conference on Web and Internet Economics, WINE 2010, 2010, pp. 246–257.

[12] M. Feldman, Maximizing symmetric submodular functions, ACM Trans. Algorithms 13 (3) (2017) 39:1–39:36.

[13] C. Chekuri, J. Vondrák, R. Zenklusen, Submodular function maximization via the multilinear relaxation and contention resolution schemes, SIAM J. Comput. 43 (6) (2014) 1831–1879.

[14] M. Feldman, J. Naor, R. Schwartz, A unified continuous greedy algorithm for submodular maximization, in: Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2011, IEEE Computer Society, 2011, pp. 570–579.

[15] A. Kulik, H. Shachnai, T. Tamir, Approximations for monotone and nonmonotone submodular maximization with knapsack constraints, Math. Oper. Res. 38 (4) (2013) 729–739.

[16] G. Călinescu, C. Chekuri, M. Pál, J. Vondrák, Maximizing a submodular set function subject to a matroid constraint, in: Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization, IPCO 2007, Vol. 4513 of LNCS, Springer, 2007, pp. 182–196.

[17] K. K. Sarpatwar, B. Schieber, H. Shachnai, Constrained submodular maximization via greedy local search, Oper. Res. Lett. 47 (1) (2019) 1–6.

[18] A. A. Schäffer, M. Yannakakis, Simple local search problems that are hard to solve, SIAM J. Comput. 20 (1) (1991) 56–87.

[19] G. Amanatidis, G. Birmpas, E. Markakis, On budget-feasible mechanism design for symmetric submodular objectives, in: Proceedings of the 13th International Conference on Web and Internet Economics, WINE 2017, 2017, pp. 1–15.