# Submodular Maximization by Simulated Annealing

Shayan Oveis Gharan* Jan Vondrák†

**Abstract**

We consider the problem of maximizing a nonnegative (possibly non-monotone) submodular set function with or without constraints. Feige et al. [9] showed a 2/5-approximation for the unconstrained problem and also proved that no approximation better than 1/2 is possible in the value oracle model. Constant-factor approximation has been also known for submodular maximization subject to a matroid independence constraint (a factor of 0.309 [33]) and for submodular maximization subject to a matroid base constraint, provided that the fractional base packing number $\nu$ is bounded away from 1 (a 1/4-approximation assuming that $\nu \geq 2$ [33]).

In this paper, we propose a new algorithm for submodular maximization which is based on the idea of *simulated annealing*. We prove that this algorithm achieves improved approximation for two problems: a 0.41-approximation for unconstrained submodular maximization, and a 0.325-approximation for submodular maximization subject to a matroid independence constraint.

On the hardness side, we show that in the value oracle model it is impossible to achieve a 0.478-approximation for submodular maximization subject to a matroid independence constraint, or a 0.394-approximation subject to a matroid base constraint in matroids with two disjoint bases. Even for the special case of cardinality constraint, we prove it is impossible to achieve a 0.491-approximation. (Previously it was conceivable that a 1/2-approximation exists for these problems.) It is still an open question whether a 1/2-approximation is possible for unconstrained submodular maximization.

## 1 Introduction

A function $f : 2^X \to \mathbb{R}$ is called *submodular* if for any $S, T \subseteq X$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. In this paper, we consider the problem of *maximizing a nonnegative submodular function*. This means, given a submodular function $f : 2^X \to \mathbb{R}_+$, find a set $S \subseteq X$ (possibly under some constraints) maximizing $f(S)$. We assume a *value oracle* access to the submodular function; i.e., for a given set $S$, the algorithm can query an oracle to find its value $f(S)$.

**Background.** Submodular functions have been studied for a long time in the context of combinatorial optimization. Lovász in his seminal paper [25] discussed various properties of submodular functions and noted that they exhibit certain properties reminiscent of convex functions - namely the fact that a naturally defined extension of a submodular function to a continuous function (the "Lovász extension") is convex. This point of view explains why submodular functions can be *minimized* efficiently [16, 11, 28].

On the other hand, submodular functions also exhibit properties closer to concavity, for example a function $f(S) = \phi(|S|)$ is submodular if and only if $\phi$ is concave. However, the problem of *maximizing* a submodular function captures problems such as Max Cut [13] and Max $k$-cover [7] which are NP-hard. Hence, we cannot expect to maximize a submodular function exactly; still, the structure of a submodular functions (in particular, the "concave aspect" of submodularity) makes it possible to achieve non-trivial results for maximization problems. Instead of the Lovász extension, the construct which turns out to be useful for maximization problems is the *multilinear extension*, introduced in [4]. This extension has been used to design an optimal $(1 - 1/e)$-approximation for the problem of maximizing a monotone submodular function subject to a matroid independence constraint [32, 5], improving the greedy 1/2-approximation of Fisher, Nemhauser and Wolsey [10]. In contrast to the Lovász extension, the multilinear extension captures the concave as well as convex aspects of submodularity. A number of improved results followed for maximizing monotone submodular functions subject to various constraints [21, 22, 23, 6].

This paper is concerned with submodular functions which are not necessarily monotone. We only assume that the function is nonnegative.[1] The problem of maximizing a nonnegative submodular function has been studied in the operations research community, with many heuristic solutions proposed: data-correcting search methods [14, 15, 20], accelatered greedy algo-

*Stanford University, Stanford, CA; shayan@stanford.edu; this work was done partly while the author was at IBM Almaden Research Center, San Jose, CA.

†IBM Almaden Research Center, San Jose, CA; jvondrak@us.ibm.com

[1] For submodular functions without any restrictions, verifying whether the maximum of the function is greater than zero or not requires exponentially many queries. Thus, there is no non-trivial multiplicative approximation for this problem.

| Problem | Prior approximation | New approximation | New hardness | Prior hardness |
|---|---|---|---|---|
| $\max\{f(S) : S \subseteq X\}$ | 0.4 | 0.41 | – | 0.5 |
| $\max\{f(S) : \|S\| \leq k\}$ | 0.309 | 0.325 | 0.491 | 0.5 |
| $\max\{f(S) : \|S\| = k\}$ | 0.25 | – | 0.491 | 0.5 |
| $\max\{f(S) : S \in \mathcal{I}\}$ | 0.309 | 0.325 | 0.478 | 0.5 |
| $\max\{f(S) : S \in \mathcal{B}\}^*$ | 0.25 | – | 0.394 | 0.5 |

Figure 1: Summary of results: $f(S)$ is nonnegative submodular, $\mathcal{I}$ denotes independent sets in a matroid, and $\mathcal{B}$ bases in a matroid. $^*$: in this line (matroid base constraint) we assume the case where the matroid contains two disjoint bases. The hardness results hold in the value oracle model.

rithms [27], and polyhedral algorithms [24]. The first algorithms with provable performace guarantees for this problem were given by Feige, Mirrokni and Vondrák [9]. They presented several algorithms achieving constant-factor approximation, the best approximation factor being 2/5 (by a randomized local search algorithm). They also proved that a better than 1/2 approximation for submodular maximization would require exponentially many queries in the value oracle model. This is true even for symmetric submodular functions, in which case a 1/2-approximation is easy to achieve [9].

Recently, approximation algorithms have been designed for nonnegative submodular maximization subject to various constraints [22, 23, 33, 17]. (Submodular minimization subject to additional constraints has been also studied [30, 12, 18].) The results most relevant to this work are that a nonnegative submodular functions can be maximized subject to a matroid independence constraint within a factor of 0.309, while a better than 1/2-approximation is impossible [33], and there is $\frac{1}{2}(1 - \frac{1}{\nu} - o(1))$-approximation subject to a matroid base constraint for matroids of fractional base packing number at least $\nu \in [1, 2]$, while a better than $(1 - \frac{1}{\nu})$-approximation in this setting is impossible [33]. For explicitly represented instances of unconstrained submodular maximization, Austrin [1] recently proved that assuming the Unique Games Conjecture, the problem is NP-hard to approximate within a factor of 0.695.

**Our results.** In this paper, we propose a new algorithm for submodular maximization, using the concept of *simulated annealing*. The main idea is to perform a local search under a certain amount of random noise which gradually decreases to zero. This helps avoid bad local optima at the beginning, and provides gradually more and more refined local search towards the end. Algorithms of this type have been widely employed for large-scale optimization problems, but they are notoriously difficult to analyze.

We prove that a simulated annealing algorithm achieves at least a 0.41-approximation for the maximization of any nonnegative submodular function without constraints, improving upon the previously known 0.4-approximation [9]. (Although our initial hope was that this algorithm might achieve a 1/2-approximation, we found an example where it achieves only a factor of $17/35 \simeq 0.486$; see subsection 3.1.) We also prove that a similar algorithm achieves a 0.325-approximation for the maximization of a nonnegative submodular function subject to a matroid independence constraint (improving the previously known factor of 0.309 [33]).

On the hardness side, we show the following results in the value oracle model: For submodular maximization under a matroid base constraint, it is impossible to achieve a 0.394-approximation even in the special case when the matroid contains two disjoint bases. For maximizing a nonnegative submodular function subject to a matroid independence constraint, we prove it is impossible to achieve a 0.478-approximation. For the special case of a cardinality constraint ($\max\{f(S) : |S| \leq k\}$ or $\max\{f(S) : |S| = k\}$), we prove a hardness threshold of 0.491. We remark that only a hardness of $(1/2 + \epsilon)$-approximation was known for all these problems prior to this work. For matroids of fractional base packing number $\nu = k/(k-1), k \in \mathbb{Z}$, we show that submodular maximization subject to a matroid base constraint does not admit a $(1 - e^{-1/k} + \epsilon)$-approximation for any $\epsilon > 0$, improving the previously known threshold of $1/k + \epsilon$ [33]. These results rely on the notion of a *symmetry gap* and the hardness construction of [33].

**Organization.** The rest of the paper is organized as follows. In Section 2, we discuss the notions of multilinear relaxation and simulated annealing, which form the basis of our algorithms. In Section 3, we describe and analyze our 0.41-approximation for unconstrained submodular maximization. In Section 4, we describe our 0.325-approximation for submodular maximization subject to a matroid independence constraint. In Section 5, we present our hardness results based on the notion of symmetry gap. We defer some technical lemmas to the appendix.

## 2 Preliminaries

Our algorithm combines the following two concepts. The first one is *multilinear relaxation*, which has recently proved to be very useful for optimization problems involving submodular functions (see [4, 32, 5, 21, 22, 33]). The second concept is *simulated annealing*, which has been used successfully by practitioners dealing with difficult optimization problems. Simulated annealing provides good results in many practical scenarios, but typically eludes rigorous analysis (with several exceptions in the literature: see e.g. [2] for general convergence results, [26, 19] for applications to volumes estimation and optimization over convex bodies, and [31, 3] for applications to counting problems).

**Multilinear relaxation.** Consider a submodular function $f : 2^X \to \mathbb{R}_+$. We define a continuous function $F : [0,1]^X \to \mathbb{R}_+$ as follows: For $\mathbf{x} \in [0,1]^X$, let $R \subseteq X$ be a random set which contains each element $i$ independently with probability $x_i$. Then we define

$$F(\mathbf{x}) := \mathbf{E}[f(R)] = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j).$$

This is the unique multilinear polynomial in $x_1, \ldots, x_n$ which coincides with $f(S)$ on the points $\mathbf{x} \in \{0,1\}^X$ (we identify such points with subsets $S \subseteq X$ in a natural way). Instead of the discrete optimization problem $\max\{f(S) : S \in \mathcal{F}\}$ where $\mathcal{F} \subseteq 2^X$ is the family of feasible sets, we consider a continuous optimization problem $\max\{F(x) : x \in P(\mathcal{F})\}$ where $P(\mathcal{F}) = \mathrm{conv}(\{\mathbf{1}_S : S \in \mathcal{F}\})$ is the polytope associated with $\mathcal{F}$. It is known due to [4, 5, 33] that any fractional solution $\mathbf{x} \in P(\mathcal{F})$ where $\mathcal{F}$ are either all subsets, or independent sets in a matroid, or matroid bases, can be rounded to an integral solution $S \in \mathcal{F}$ such that $f(S) \geq F(\mathbf{x})$. Our algorithm can be seen as a new way of approximately solving the relaxed problem $\max\{F(\mathbf{x}) : \mathbf{x} \in P(\mathcal{F})\}$.

**Simulated annealing.** The idea of simulated annealing comes from physical processes such as gradual cooling of molten metals, whose goal is to achieve the state of lowest possible energy. The process starts at a high temperature and gradually cools down to a "frozen state". The main idea behind gradual cooling is that while it is natural for a physical system to seek a state of minimum energy, this is true only in a local sense - the system does not have any knowledge of the global structure of the search space. Thus a low-temperature system would simply find a local optimum and get stuck there, which might be suboptimal. Starting the process at a high temperature means that there is more randomness in the behavior of the system. This gives the system more freedom to explore the search space, escape from bad local optima, and converge faster to a better solution. We pursue a similar strategy here.

We should remark that our algorithm is somewhat different from a direct interpretation of simulated annealing. In simulated annealing, the system would typically evolve as a random walk, with sensitivity to the objective function depending on the current temperature. Here, we adopt a simplistic interpretation of temperature as follows. Given a set $A \subset X$ and $t \in [0,1]$, we define a random set $R_t(A)$ by starting from $A$ and adding/removing each element independently with probability $t$. Instead of the objective function evaluated on $A$, we consider the expectation over $R_t(A)$. This corresponds to the *noise operator* used in the analysis of boolean functions, which was implicitly also used in the 2/5-approximation algorithm of [9]. Observe that $\mathbf{E}[f(R_t(A))] = F((1-t)\mathbf{1}_A + t\mathbf{1}_{\overline{A}})$, where $F$ is the multilinear extension of $f$. The new idea here is that the parameter $t$ plays a role similar to temperature - e.g., $t = 1/2$ means that $R_t(A)$ is uniformly random regardless of $A$ ("infinite temperature" in physics), while $t = 0$ means that there are no fluctuations present at all ("absolute zero").

We use this interpretation to design an algorithm inspired by simulated annealing: Starting from $t = 1/2$, we perform local search on $A$ in order to maximize $\mathbf{E}[f(R_t(A))]$. Note that for $t = 1/2$ this function does not depend on $A$ at all, and hence any solution is a local optimum. Then we start gradually decreasing $t$, while simultaneously running a local search with respect to $\mathbf{E}[f(R_t(A))]$. Eventually, we reach $t = 0$ where the algorithm degenerates to a traditional local search and returns an (approximate) local optimum.

We emphasize that we maintain the solution generated by previous stages of the algorithm, as opposed to running a separate local search for each value of $t$. This is also used in the analysis, whose main point is to estimate how the solution improves as a function of $t$. It is not a coincidence that the approximation provided by our algorithm is a (slight) improvement over previous algorithms. Our algorithm can be viewed as a dynamic process which at each fixed temperature $t$ corresponds to a certain variant of the algorithm from [9]. We prove that the performance of the simulated annealing process is described by a differential equation, whose initial condition can be related to the performance of a previously known algorithm. Hence the fact that an improvement can be achieved follows from the fact that the differential equation yields a positive drift at the initial point. The exact quantitative improvement depends on the solution of the differential equation, which we also present in this work.

**Notation.** In this paper, we denote vectors consistently in boldface: for example $\mathbf{x}, \mathbf{y} \in [0,1]^n$. The coordinates of $\mathbf{x}$ are denoted by $x_1, \ldots, x_n$. Subscripts next to a boldface symbol, such as $\mathbf{x}_0, \mathbf{x}_1$, denote different vectors. In particular, we use the notation $\mathbf{x}_p(A)$ to denote a vector with coordinates $x_i = p$ for $i \in A$ and $x_i = 1-p$ for $i \notin A$. In addition, we use the following notation to denote the value of certain fractional solutions:

| | $C$ | $\overline{C}$ |
|---|---|---|
| $A$ | $p$ | $p'$ |
| $B$ | $q$ | $q'$ |

$:= F(p\mathbf{1}_{A\cap C} + p'\mathbf{1}_{A\setminus C} + q\mathbf{1}_{B\cap C} + q'\mathbf{1}_{B\setminus C})$.

For example, if $p = p'$ and $q = q' = 1-p$, the diagram would represent $F(\mathbf{x}_p(A))$. Typically, $A$ will be our current solution, and $C$ an optimal solution. Later we omit the symbols $A, B, C, \overline{C}$ from the diagram.

## 3 Unconstrained Submodular Maximization

Let us describe our algorithm for unconstrained submodular maximization. We use a parameter $p \in [\frac{1}{2}, 1]$, which is related to the "temperature" discussed above by $p = 1 - t$. We also use a fixed discretization parameter $\delta = 1/n^3$.

**Algorithm 1** Simulated Annealing Algorithm For Submodular Maximization

**Input:** A submodular function $f : 2^X \to \mathbb{R}_+$.
**Output:** A subset $A \subseteq X$ satisfying $f(A) \geq 0.41 \cdot \max\{f(S) : S \subseteq X\}$.

1: **Define** $\mathbf{x}_p(A) = p\mathbf{1}_A + (1-p)\mathbf{1}_{\overline{A}}$.
2: $A \leftarrow \emptyset$.
3: **for** $p \leftarrow 1/2;\ p \leq 1;\ p \leftarrow p + \delta$ **do**
4: **while** there exists $i \in X$ such that $F(\mathbf{x}_p(A\Delta\{i\})) > F(\mathbf{x}_p(A))$ **do**
5: $A \leftarrow A\Delta\{i\}$
6: **end while**
7: **end for**
8: **return** the best solution among all sets $A$ and $\overline{A}$ encountered by the algorithm.

We remark that this algorithm would not run in polynomial time, due to the complexity of finding a local optimum in Step 4-6. This can be fixed by standard techniques (as in [9, 22, 23, 33]), by stopping when the conditions of local optimality are satisfied with sufficient accuracy. We also assume that we can evaluate the multilinear extension $F$, which can be done within a certain desired accuracy by random sampling. Since the analysis of the algorithm is already quite technical, we ignore these issues in this extended abstract and assume instead that a true local optimum is found in Step 4-6.

THEOREM 3.1. *For any submodular function $f : 2^X \to \mathbb{R}_+$, Algorithm 1 returns with high probability a solution of value at least $0.41 \cdot OPT$ where $OPT = \max_{S \subseteq X} f(S)$.*

In Theorem 3.2 we also show that Algorithm 1 does not achieve any factor better than $17/35 \simeq 0.486$. First, let us give an overview of our approach and compare it to the analysis of the 2/5-approximation in [9]. The algorithm of [9] can be viewed in our framework as follows: for a fixed value of $p$, it performs local search over points of the form $\mathbf{x}_p(A)$, with respect to element swaps in $A$, and returns a locally optimal solution. Using the conditions of local optimality, $F(\mathbf{x}_p(A))$ can be compared to the global optimum. Here, we observe the following additional property of a local optimum. If $\mathbf{x}_p(A)$ is a local optimum with respect to element swaps in $A$, then slightly increasing $p$ cannot decrease the value of $F(\mathbf{x}_p(A))$. During the local search stage, the value cannot decrease either, so in fact the value of $F(\mathbf{x}_p(A))$ is non-decreasing throughout the algorithm. Moreover, we can derive bounds on $\frac{\partial}{\partial p} F(\mathbf{x}_p(A))$ depending on the value of the current solution. Consequently, unless the current solution is already valuable enough, we can conclude that an improvement can be achieved by increasing $p$. This leads to a differential equation whose solution implies Theorem 3.1.

We proceed slowly and first prove the basic fact that if $\mathbf{x}_p(A)$ is a local optimum for a fixed $p$, we cannot lose by increasing $p$ slightly. This is intuitive, because the gradient $\nabla F$ at $\mathbf{x}_p(A)$ must be pointing away from the center of the cube $[0,1]^X$, or else we could gain by a local step.

LEMMA 3.1. *Let $p \in [\frac{1}{2}, 1]$ and suppose $\mathbf{x}_p(A)$ is a local optimum in the sense that $F(\mathbf{x}_p(A\Delta\{i\})) \leq F(\mathbf{x}_p(A))$ for all $i$. Then*

- *$\frac{\partial F}{\partial x_i} \geq 0$ if $i \in A$, and $\frac{\partial F}{\partial x_i} \leq 0$ if $i \notin A$,*
- *$\frac{\partial}{\partial p} F(\mathbf{x}_p(A)) = \sum_{i \in A} \frac{\partial F}{\partial x_i} - \sum_{i \notin A} \frac{\partial F}{\partial x_i} \geq 0$.*

*Proof.* We assume that flipping the membership of element $i$ in $A$ can only decrease the value of $F(\mathbf{x}_p(A))$. The effect of this local step on $\mathbf{x}_p(A)$ is that the value of the $i$-th coordinate changes from $p$ to $1-p$ or vice versa (depending on whether $i$ is in $A$ or not). Since $F$ is linear when only one coordinate is being changed, this implies $\frac{\partial F}{\partial x_i} \geq 0$ if $i \in A$, and $\frac{\partial F}{\partial x_i} \leq 0$ if $i \notin A$. By the chain rule, we have

$$\frac{\partial F(\mathbf{x}_p(A))}{\partial p} = \sum_{i=1}^{n} \frac{\partial F}{\partial x_i} \frac{d(\mathbf{x}_p(A))_i}{dp}.$$

Since $(\mathbf{x}_p(A))_i = p$ if $i \in A$ and $1-p$ otherwise, we get $\frac{\partial F(\mathbf{x}_p(A))}{\partial p} = \sum_{i \in A} \frac{\partial F}{\partial x_i} - \sum_{i \notin A} \frac{\partial F}{\partial x_i} \geq 0$ using the conditions above. □

In the next lemma, we prove a stronger bound on the derivative $\frac{\partial}{\partial p}F(\mathbf{x}_p(A))$ which will be our main tool in proving Theorem 3.1. This can be combined with the analysis of [9] to achieve a certain improvement. For instance, [9] implies that if $A$ is a local optimum for $p = 2/3$, we have either $f(\overline{A}) \geq \frac{2}{5}OPT$, or $F(\mathbf{x}_p(A)) \geq \frac{2}{5}OPT$. Suppose we start our analysis from the point $p = 2/3$. (The algorithm does not need to be modified, since at $p = 2/3$ it finds a local optimum in any case, and this is sufficient for the analysis.) We have either $f(\overline{A}) > \frac{2}{5}OPT$ or $F(\mathbf{x}_p(A)) > \frac{2}{5}OPT$, or else by Lemma 3.2, $\frac{\partial}{\partial p}F(\mathbf{x}_p(A))$ is a constant fraction of $OPT$:

$$\frac{1}{3}\cdot\frac{\partial}{\partial p}F(\mathbf{x}_p(A)) \geq OPT\left(1-\frac{4}{5}-\frac{1}{3}\times\frac{2}{5}\right) = \frac{1}{15}OPT.$$

Therefore, in some $\delta$-sized interval, the value of $F(\mathbf{x}_p(A))$ will increase at a slope proportional to $OPT$. Thus the approximation factor of Algorithm 1 is strictly greater than 2/5. We remark that we use a different starting point to achieve the factor of 0.41.

The key lemma in our analysis states the following.

LEMMA 3.2. *Let $OPT = \max_{S\subseteq X} f(S)$, $p \in [\frac{1}{2}, 1]$ and suppose $\mathbf{x}_p(A)$ is a local optimum in the sense that $F(\mathbf{x}_p(A\Delta\{i\})) \leq F(\mathbf{x}_p(A))$ for all $i$. Then*

$$(1-p)\cdot\frac{\partial}{\partial p}F(\mathbf{x}_p(A)) \geq OPT - 2F(\mathbf{x}_p(A)) - (2p-1)f(\overline{A}).$$

*Proof.* Let $C$ denote an optimal solution, i.e. $f(C) = OPT$. Let $A$ denote a local optimum with respect to $F(\mathbf{x}_p(A))$, and $B = \overline{A}$ its complement. In our notation using diagrams,

$$F(\mathbf{x}_p(A)) = F(p\mathbf{1}_A + (1-p)\mathbf{1}_B) = \begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline\end{array}$$

The top row is the current solution $A$, the bottom row is its complement $B$, and the left-hand column is the optimum $C$. We proceed in two steps. Define

$$G(\mathbf{x}) = (\mathbf{1}_C - \mathbf{x})\cdot\nabla F(\mathbf{x}) = \sum_{i\in C}(1-x_i)\frac{\partial F}{\partial x_i} - \sum_{i\notin C} x_i\frac{\partial F}{\partial x_i}$$

to denote the derivative of $F$ when moving from $\mathbf{x}$ towards the actual optimum $\mathbf{1}_C$. By Lemma 3.1, we have

$$\begin{aligned}(1-p)\frac{\partial F(\mathbf{x}_p(A))}{\partial p} &= (1-p)\left(\sum_{i\in A}\frac{\partial F}{\partial x_i} - \sum_{i\in B}\frac{\partial F}{\partial x_i}\right)\\ &\geq (1-p)\left(\sum_{i\in A\cap C}\frac{\partial F}{\partial x_i} - \sum_{i\in B\setminus C}\frac{\partial F}{\partial x_i}\right)\\ &\quad - p\left(\sum_{i\in A\setminus C}\frac{\partial F}{\partial x_i} - \sum_{i\in B\cap C}\frac{\partial F}{\partial x_i}\right) = G(\mathbf{x}_p(A))\end{aligned}$$

using the definition of $\mathbf{x}_p(A)$ and the fact that $\frac{\partial F}{\partial x_i} \geq 0$ for $i \in A\setminus C$ and $\frac{\partial F}{\partial x_i} \leq 0$ for $i \in B\cap C$.

Next, we use Lemma A.1 to estimate $G(\mathbf{x}_p(A))$ as follows. To simplify notation, we denote $\mathbf{x}_p(A)$ simply by $\mathbf{x}$. If we start from $\mathbf{x}$ and increase the coordinates in $A\cap C$ by $(1-p)$ and those in $B\cap C$ by $p$, Lemma A.1 says the value of $F$ will change by

$$\begin{aligned}&\begin{array}{|c|c|}\hline 1 & \text{p} \\ \hline 1 & \text{1-p} \\ \hline\end{array} - \begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline\end{array}\\ &= F(\mathbf{x} + (1-p)\mathbf{1}_{A\cap C} + p\mathbf{1}_{B\cap C}) - F(\mathbf{x})\\ &\leq (1-p)\sum_{i\in A\cap C}\frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} + p\sum_{i\in B\cap C}\frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}}. \qquad (3.1)\end{aligned}$$

Similarly, if we decrease the coordinates in $A\setminus C$ by $p$ and those in $B\setminus C$ by $1-p$, the value will change by

$$\begin{aligned}&\begin{array}{|c|c|}\hline \text{p} & 0 \\ \hline \text{1-p} & 0 \\ \hline\end{array} - \begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline\end{array}\\ &= F(\mathbf{x} - p\mathbf{1}_{A\setminus C} - (1-p)\mathbf{1}_{B\setminus C}) - F(\mathbf{x})\\ &\leq -(1-p)\sum_{i\in B\setminus C}\frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} - p\sum_{i\in A\setminus C}\frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}}. \qquad (3.2)\end{aligned}$$

Adding inequalities(3.1), (3.2) and noting the expression for $G(\mathbf{x})$ above, we obtain:

$$\begin{array}{|c|c|}\hline 1 & \text{p} \\ \hline 1 & \text{1-p} \\ \hline\end{array} + \begin{array}{|c|c|}\hline \text{p} & 0 \\ \hline \text{1-p} & 0 \\ \hline\end{array} - 2\begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline\end{array} \leq G(\mathbf{x}). \quad (3.3)$$

It remains to relate the LHS of equation (3.3) to the value of $OPT$. We use the "threshold lemma" (see Lemma A.3, and the accompanying example with equation (A.1)):

$$\begin{aligned}\begin{array}{|c|c|}\hline \text{p} & 0 \\ \hline \text{1-p} & 0 \\ \hline\end{array} &\geq (1-p)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array} + (2p-1)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline\end{array}\\ &\quad + (1-p)\begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline\end{array}\\ &\geq (1-p)OPT + (2p-1)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline\end{array},\\ \begin{array}{|c|c|}\hline 1 & \text{p} \\ \hline 1 & \text{1-p} \\ \hline\end{array} &\geq (1-p)\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline\end{array} + (2p-1)\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline\end{array}\\ &\quad + (1-p)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline\end{array}\\ &\geq (2p-1)\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline\end{array} + (1-p)OPT.\end{aligned}$$

Combining these inequalities with (3.3), we get

$$\begin{aligned}G(\mathbf{x}) \geq\ & 2(1-p)OPT - 2\begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline\end{array}\\ &+ (2p-1)\left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline\end{array} + \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline\end{array}\right].\end{aligned}$$

Recall that $F(\mathbf{x}) = \begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline \end{array}$. Finally, we add $(2p-1)f(\overline{A}) = (2p-1)\begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}$ to this inequality, so that we can use submodularity to take advantage of the last two terms:

$$\begin{aligned} G(\mathbf{x})+(2p-1)f(\overline{A}) \geq\ & 2(1-p)OPT - 2\begin{array}{|c|c|}\hline \text{p} & \text{p} \\ \hline \text{1-p} & \text{1-p} \\ \hline \end{array} \\ & +(2p-1)\left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 1 & 1 \\ \hline \end{array}\right] \\ \geq & 2(1-p)OPT - 2F(\mathbf{x}_p(A)) + (2p-1)OPT \\ = & OPT - 2F(\mathbf{x}_p(A)). \end{aligned}$$

□

We have proved that unless the current solution is already very valuable, there is a certain improvement that can be achieved by increasing $p$. The next lemma transforms this statement into an inequality describing the evolution of the simulated-annealing algorithm.

LEMMA 3.3. *Let $A(p)$ denote the local optimum found by the simulated annealing algorithm at temperature $t = 1-p$, and let $\Phi(p) = F(\mathbf{x}_p(A(p)))$ denote its value. Assume also that for all $p$, we have $f(\overline{A(p)}) \leq \beta$. Then*

$$\frac{1-p}{\delta}(\Phi(p+\delta) - \Phi(p)) \ \geq\ (1-2\delta n^2)OPT - 2\Phi(p) - (2p-1)\beta.$$

*Proof.* Here we combine the positive drift obtained from decreasing the temperature (described by Lemma 3.2) and from local search (which is certainly nonnegative). Consider the local optimum $A$ obtained at temperature $t = 1-p$. Its value is $\Phi(p) = F(\mathbf{x}_p(A))$. By decreasing temperature by $\delta$, we obtain a solution $\mathbf{x}_{p+\delta}(A)$, whose value can be estimated in the first order by the derivative at $p$ (see Lemma A.2 for a precise argument):

$$F(\mathbf{x}_{p+\delta}(A)) \geq F(\mathbf{x}_p(A)) + \delta\frac{\partial F(\mathbf{x}_p(A))}{\partial p} - \delta^2 n^2 \sup\left|\frac{\partial^2 F}{\partial x_i \partial x_j}\right|.$$

This is followed by another local-search stage, in which we obtain a new local optimum $A'$. In this stage, the value of the objective function cannot decrease, so we have $\Phi(p+\delta) = F(\mathbf{x}_{p+\delta}(A')) \geq F(\mathbf{x}_{p+\delta}(A))$. We have $\sup|\frac{\partial^2 F}{\partial x_i \partial x_j}| \leq \max_{S,i,j}|f(S+i+j) - f(S+i) - f(S+j) + f(S)| \leq 2OPT$. We also estimate $\frac{\partial}{\partial p}F(\mathbf{x}_p(A))$ using Lemma 3.1, to obtain

$$\begin{aligned} \Phi(p+\delta) \geq & F(\mathbf{x}_{p+\delta}(A)) \\ \geq & F(\mathbf{x}_p(A)) - 2\delta^2 n^2\, OPT \\ & + \frac{\delta}{1-p}(OPT - 2F(\mathbf{x}_p(A)) - (2p-1)f(\overline{A})). \end{aligned}$$

Finally, we use $f(\overline{A}) \leq \beta$ and $F(\mathbf{x}_p(A)) = \Phi(p)$ to derive the statement of the lemma. □

By taking $\delta \to 0$, the statement of Lemma 3.3 leads naturally to the following differential equation:

$$(1-p)\Phi'(p) \geq OPT - 2\Phi(p) - (2p-1)\beta. \qquad (3.4)$$

We assume here that $\delta$ is so small that the difference between the solution of this differential inequality and the actual behavior of our algorithm is negligible. (We could replace $OPT$ by $(1-\epsilon)OPT$, carry out the analysis and then let $\epsilon \to 0$; however, we shall spare the reader of this annoyance.)

Our next step is to solve this differential equation, given certain initial conditions. Without loss of generality, we assume that $OPT = 1$.

LEMMA 3.4. *Assume that $OPT = 1$. Let $\Phi(p)$ denote the value of the solution at temperature $t = 1-p$. Assume that $\Phi(p_0) = v_0$ for some $p_0 \in (\frac{1}{2}, 1)$, and $f(\overline{A(p)}) \leq \beta$ for all $p$. Then for any $p \in (p_0, 1)$,*

$$\Phi(p) \geq \frac{1}{2}(1-\beta) + 2\beta(1-p) - \frac{(1-p)^2}{(1-p_0)^2}\left(\frac{1}{2}(1-\beta) + 2\beta(1-p_0) - v_0\right).$$

*Proof.* We rewrite Equation (3.4) using the following trick:

$$\begin{aligned} (1-p)^3\frac{d}{dp}((1-p)^{-2}\Phi(p)) = & (1-p)^3(2(1-p)^{-3}\Phi(p) \\ & + (1-p)^{-2}\Phi'(p)) \\ = & 2\Phi(p) + (1-p)\Phi'(p). \end{aligned}$$

Therefore, Lemma 3.3 states that

$$\begin{aligned} (1-p)^3\frac{d}{dp}(p^{-2}\Phi(p)) \ &\geq\ OPT - (2p-1)\beta \\ &=\ 1 - \beta + 2\beta(1-p) \end{aligned}$$

which is equivalent to

$$\frac{d}{dp}((1-p)^{-2}\Phi(p)) \geq \frac{1-\beta}{(1-p)^3} + \frac{2\beta}{(1-p)^2}.$$

For any $p \in (p_0, 1)$, the fundamental theorem of calculus implies that

$$\begin{aligned}(1-p)^{-2}\Phi(p)&-(1-p_0)^{-2}\Phi(p_0)\\ &\geq \int_{p_0}^{p}\left(\frac{1-\beta}{(1-\tau)^3}+\frac{2\beta}{(1-\tau)^2}\right)d\tau\\ &=\left[\frac{1-\beta}{2(1-\tau)^2}+\frac{2\beta}{1-\tau}\right]_{p_0}^{p}\\ &=\frac{1-\beta}{2(1-p)^2}+\frac{2\beta}{1-p}-\frac{1-\beta}{2(1-p_0)^2}-\frac{2\beta}{1-p_0}.\end{aligned}$$

Multiplying by $(1-p)^2$, we obtain

$$\begin{aligned}\Phi(p) \geq&\frac{1}{2}(1-\beta)+2\beta(1-p)\\ &+\frac{(1-p)^2}{(1-p_0)^2}\left(\Phi(p_0)-\frac{1}{2}(1-\beta)-2\beta(1-p_0)\right).\end{aligned}$$

□

In order to use this lemma, recall that the parameter $\beta$ is an upper bound on the values of $f(\overline{A})$ throughout the algorithm. This means that we can choose $\beta$ to be our "target value": if $f(\overline{A})$ achieves value more than $\beta$ at some point, we are done. If $f(\overline{A})$ is always upper-bounded by $\beta$, we can use Lemma 3.4, hopefully concluding that for some $p$ we must have $\Phi(p) \geq \beta$.

In addition, we need to choose a suitable initial condition. As a first attempt, we can try to plug in $p_0 = 1/2$ and $v_0 = 1/4$ as a starting point (the uniformly random 1/4-approximation provided by [9]). We would obtain

$$\Phi(p) \geq \frac{1}{2}(1-\beta)+2\beta(1-p)-(1+2\beta)(1-p)^2.$$

However, this is not good enough. For example, if we choose $\beta = 2/5$ as our target value, we obtain $\Phi(p) \geq \frac{3}{10}+\frac{4}{5}(1-p)-\frac{9}{5}(1-p)^2$. It can be verified that this function stays strictly below $2/5$ for all $p \in [\frac{1}{2}, 1]$. So this does not even match the performance of the 2/5-approximation of [9].

As a second attempt, we can use the 2/5-approximation itself as a starting point. The analysis of [9] implies that if $A$ is a local optimum for $p_0 = 2/3$, we have either $f(\overline{A}) \geq 2/5$, or $F(\mathbf{x}_p(A)) \geq 2/5$. This means that we can use the starting point $p_0 = 2/3, v_0 = 2/5$ with a target value of $\beta = 2/5$ (effectively ignoring the behavior of the algorithm for $p < 2/3$). Lemma 3.4 gives

$$\Phi(p) \geq \frac{3}{10}+\frac{4}{5}(1-p)-\frac{3}{2}(1-p)^2.$$

The maximum of this function is attained at $p_0 = 11/15$ which gives $\Phi(p_0) \geq 61/150 > 2/5$. This is a good sign - however, it does not imply that the algorithm actually achieves a 61/150-approximation, because we have used $\beta = 2/5$ as our target value. (Also, note that $61/150 < 0.41$, so this is not the way we achieve our main result.)

In order to get an approximation guarantee better than 2/5, we need to revisit the analysis of [9] and compute the approximation factor of a local optimum as a function of the temperature $t = 1-p$ and the complementary solution $f(\overline{A}) = \beta$.

LEMMA 3.5. *Assume $OPT = 1$. Let $q \in [\frac{1}{3}, \frac{1}{1+\sqrt{2}}]$, $p = 1-q$ and let $A$ be a local optimum with respect to $F(\mathbf{x}_p(A))$. Let $\beta = f(\overline{A})$. Then*

$$F(\mathbf{x}_p(A)) \geq \frac{1}{2}(1-q^2)-q(1-2q)\beta.$$

*Proof.* $A$ is a local optimum with respect to the objective function $F(\mathbf{x}_p(A))$. We denote $\mathbf{x}_p(A)$ simply by $\mathbf{x}$. Let $C$ be a global optimum and $B = \overline{A}$. As we argued in the proof of Lemma 3.2, we have

$$\begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array} \geq \begin{array}{|c|c|}\hline \mathrm{p} & 0\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array}$$

and also

$$\begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array} \geq \begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline 1 & \mathrm{q}\\ \hline\end{array}$$

We apply Lemma A.4 which states that $F(\mathbf{x}) \geq \mathbf{E}\left[f((T_{>\lambda_1}(\mathbf{x}) \cap C) \cup (T_{>\lambda_2}(\mathbf{x}) \setminus C))\right]$, where $\lambda_1, \lambda_2$ are independent and uniformly random in $[0,1]$. This yields the following (after dropping some terms which are nonnegative):

$$\begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array} \geq \begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline 1 & \mathrm{q}\\ \hline\end{array} \geq pq \begin{array}{|c|c|}\hline 1 & 0\\ \hline 1 & 0\\ \hline\end{array} + p(p-q) \begin{array}{|c|c|}\hline 1 & 0\\ \hline 0 & 0\\ \hline\end{array} + q^2 \begin{array}{|c|c|}\hline 1 & 0\\ \hline 1 & 1\\ \hline\end{array} + (p-q)q \begin{array}{|c|c|}\hline 1 & 0\\ \hline 0 & 1\\ \hline\end{array} \tag{3.5}$$

$$\begin{array}{|c|c|}\hline \mathrm{p} & \mathrm{p}\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array} \geq \begin{array}{|c|c|}\hline \mathrm{p} & 0\\ \hline \mathrm{q} & \mathrm{q}\\ \hline\end{array} \geq pq \begin{array}{|c|c|}\hline 1 & 0\\ \hline 1 & 0\\ \hline\end{array} + p(p-q) \begin{array}{|c|c|}\hline 1 & 1\\ \hline 1 & 0\\ \hline\end{array} + q^2 \begin{array}{|c|c|}\hline 0 & 0\\ \hline 1 & 0\\ \hline\end{array} + (p-q)q \begin{array}{|c|c|}\hline 0 & 1\\ \hline 1 & 0\\ \hline\end{array} \tag{3.6}$$

The first term in each bound is $pq \cdot OPT$. However, to make use of the remaining terms, we must add some terms on both sides. The terms we add are $\frac{1}{2}(-p^3+p^2q+2pq^2)f(A)+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)f(B)$; it can be verified that both coefficients are nonnegative for $q \in [\frac{1}{3}, \frac{1}{1+\sqrt{2}}]$. Also, the coefficients are chosen so that they sum up to $p^2q-q^3 = q(p^2-q^2) = q(p-q)$,

the coefficient in front of the last term in each equation. Using submodularity, we get

$$\begin{aligned}
&\frac{1}{2}(-p^3+p^2q+2pq^2)\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array}+(p-q)q\begin{array}{|c|c|}\hline 1&0\\\hline 0&1\\\hline\end{array}\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline\end{array}\\
=&\frac{1}{2}(-p^3+p^2q+2pq^2)\left[\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 0&1\\\hline\end{array}\right] \qquad (3.7)\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\left[\begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 0&1\\\hline\end{array}\right]\\
\geq&\frac{1}{2}(-p^3+p^2q+2pq^2)\begin{array}{|c|c|}\hline 1&0\\\hline 0&0\\\hline\end{array}\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\begin{array}{|c|c|}\hline 1&0\\\hline 1&1\\\hline\end{array}. \qquad (3.8)
\end{aligned}$$

Similarly, we get

$$\begin{aligned}
&\frac{1}{2}(-p^3+p^2q+2pq^2)\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array}+(p-q)q\begin{array}{|c|c|}\hline 0&1\\\hline 1&0\\\hline\end{array}\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline\end{array}\\
=&\frac{1}{2}(-p^3+p^2q+2pq^2)\left[\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 0&1\\\hline 1&0\\\hline\end{array}\right]\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\left[\begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline\end{array}+\begin{array}{|c|c|}\hline 0&1\\\hline 1&0\\\hline\end{array}\right]\\
\geq&\frac{1}{2}(-p^3+p^2q+2pq^2)\begin{array}{|c|c|}\hline 1&1\\\hline 1&0\\\hline\end{array}\\
&\quad+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3)\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array}. \qquad (3.9)
\end{aligned}$$

Putting equations (3.5), (3.6) (3.8) and (3.9) all together, we get

$$\begin{aligned}
&2\begin{array}{|c|c|}\hline p&p\\\hline q&q\\\hline\end{array}+(-p^3+p^2q+2pq^2)\begin{array}{|c|c|}\hline 1&1\\\hline 0&0\\\hline\end{array}\\
&+(p^3+p^2q-2pq^2-2q^3)\begin{array}{|c|c|}\hline 0&0\\\hline 1&1\\\hline\end{array}\\
&\quad\geq 2pq\begin{array}{|c|c|}\hline 1&0\\\hline 1&0\\\hline\end{array}\\
&+(p^2-pq+\frac{1}{2}(-p^3+p^2q+2pq^2))\left[\begin{array}{|c|c|}\hline 1&1\\\hline 1&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 0&0\\\hline\end{array}\right]\\
&+(q^2+\frac{1}{2}(p^3+p^2q-2pq^2-2q^3))\left[\begin{array}{|c|c|}\hline 1&0\\\hline 1&1\\\hline\end{array}+\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array}\right]\\
&\quad=2pq\begin{array}{|c|c|}\hline 1&0\\\hline 1&0\\\hline\end{array}\\
&+\frac{1}{2}p^2\left[\begin{array}{|c|c|}\hline 1&1\\\hline 1&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 0&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 1&1\\\hline\end{array}+\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array}\right].
\end{aligned}$$

where the simplification came about by using the elementary relations $p(p-q)=p(p-q)(p+q)=p(p^2-q^2)$ and $q^2=q^2(p+q)$. Submodularity implies

$$\begin{array}{|c|c|}\hline 1&0\\\hline 0&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 0&0\\\hline 1&0\\\hline\end{array}\geq\begin{array}{|c|c|}\hline 1&0\\\hline 1&0\\\hline\end{array}=OPT$$

and

$$\begin{array}{|c|c|}\hline 1&1\\\hline 1&0\\\hline\end{array}+\begin{array}{|c|c|}\hline 1&0\\\hline 1&1\\\hline\end{array}\geq\begin{array}{|c|c|}\hline 1&0\\\hline 1&0\\\hline\end{array}=OPT,$$

so we get, replacing the respective diagrams by $F(\mathbf{x})$, $f(A)$ and $f(B)$,

$$\begin{aligned}
&2\ F(\mathbf{x})+(-p^3+p^2q+2pq^2)f(A)+(p^3+p^2q-2pq^2-2q^3)f(B)\\
&\geq(2pq+p^2)OPT=(1-q^2)OPT
\end{aligned}$$

again using $(p+q)^2=1$. Finally, we assume that $f(A)\leq\beta$ and $f(B)\leq\beta$, which means

$$\begin{aligned}
2F(\mathbf{x}) &\geq (1-q^2)OPT-(2p^2q-2q^3)\beta\\
&= (1-q^2)OPT-2q(p-q)\beta\\
&= (1-q^2)OPT-2q(1-2q)\beta.
\end{aligned}$$

□

Now we can finally prove Theorem 3.1. Consider Lemma 3.4. Starting from $\Phi(p_0)=v_0$, we obtain the following bound for any $p\in(p_0,1)$:

$$\begin{aligned}
\Phi(p)\geq&\frac{1}{2}(1-\beta)+2\beta(1-p)\\
&-\frac{(1-p)^2}{(1-p_0)^2}\left(\frac{1}{2}(1-\beta)+2\beta(1-p_0)-v_0\right).
\end{aligned}$$

By optimizing this quadratic function, we obtain that the maximum is attained at $p_1=\frac{\beta(1-p_0)^2}{(1-\beta)/2+2\beta(1-p_0)-v_0}$ and the corresponding bound is

$$\Phi(p_1)\geq\frac{1-\beta}{2}+\frac{\beta^2(1-p_0)^2}{\frac{1}{2}(1-\beta)+2\beta(1-p_0)-v_0}.$$

Lemma 3.5 implies that a local optimum at temperature $q=1-p_0\in[\frac{1}{3},\frac{1}{1+\sqrt{2}}]$ has value $v_0\geq\frac{1}{2}(1-q^2)-q(1-2q)\beta=p_0-\frac{1}{2}p_0^2-(1-p_0)(2p_0-1)\beta$. Therefore, we obtain

$$\Phi(p_1)\geq\frac{1-\beta}{2}+\frac{\beta^2(1-p_0)^2}{\frac{1}{2}(1-\beta)+2\beta(1-p_0)-p_0+\frac{1}{2}p_0^2+(1-p_0)(2p_0-1)\beta}.$$

We choose $p_0=\frac{\sqrt{2}}{1+\sqrt{2}}$ and solve for a value of $\beta$ such that $\Phi(p_1)\geq\beta$. This value can be found as a solution of a quadratic equation and is equal to

$$\beta=\frac{1}{401}\left(37+22\sqrt{2}+(30\sqrt{2}+14)\sqrt{-5\sqrt{2}+10}\right).$$

It can be verified that $\beta>0.41$. This completes the proof of Theorem 3.1.

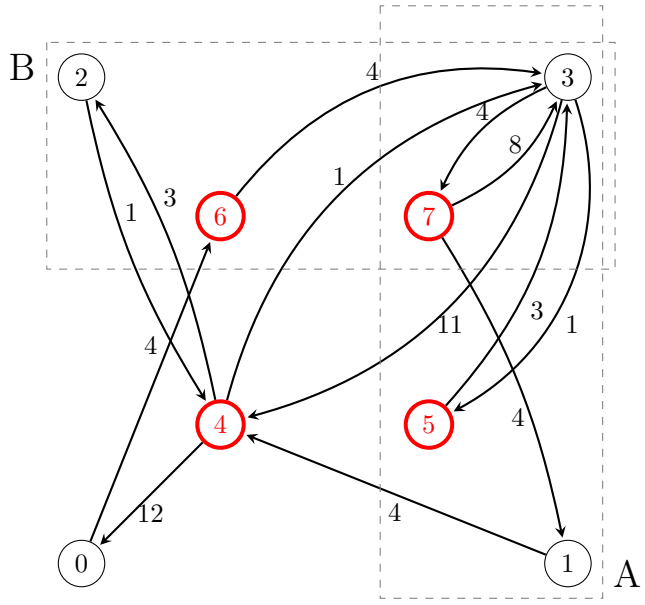


Figure 2: Hard instance of the unconstrained submodular maximization problem, where Algorithm 1 may get value no more than 17. The bold vertices $\{4,5,6,7\}$ represent the optimum set with value $OPT = 35$.

**3.1 Upper bound on the performance of the simulated annealing algorithm.** In this section we show that the simulated annealing algorithm 1 for unconstrained submodular maximization does not give a $\frac{1}{2}$-approximation even on instances of the directed maximum cut problem. We provide a directed graph $G$ (found by an LP-solver) and a set of local optimums for all values of $p \in [\frac{1}{2}, 1]$, such the value of $f$ on each of them or their complement is at most 0.486 of $OPT$.

THEOREM 3.2. *There exists an instance of the unconstrained submodular maximization problem, such that the approximation factor of Algorithm 1 is* $17/35 < 0.486$.

*Proof.* Let $f$ be the cut function of the directed graph $G$ in Figure 2. We show that the set $A = \{1,3,5,7\}$ is a local optimum for all $p \in [\frac{1}{2}, \frac{3}{4}]$ and the set $B = \{2,4,6,8\}$ is a local optimum for all $p \in [\frac{3}{4}, 1]$. Moreover, since we have $F(\mathbf{x}_{3/4}(A)) = F(\mathbf{x}_{3/4}(B) = 16.25$, it is possible that in a run of the simulated annealing algorithm 1, the set $A$ is chosen and remains as a local optimum fort $p = 1/2$ to $p = 3/4$. Then the local optimum changes to $B$ and remains until the end of the algorithm. If the algorithm follows this path then its approximation ratio is $17/35$. This is because the value of the optimum set $f(\{4,5,6,7\}) = 35$, while $\max\{f(A), f(B), f(\overline{A}), f(\overline{B})\} = 17$. We remark that even sampling from $A, \overline{A}$ (or from $B, \overline{B}$) with probabilities $p, q$ does not give value more than 17.

It remains to show that the set $A$ is in fact a local optimum for all $p \in [\frac{1}{2}, \frac{3}{4}]$. We just need to show that all the elements in $A$ have a non-negative partial derivative and the elements in $\overline{A}$ have a non-positive partial derivative. Let $p \in [\frac{1}{2}, \frac{3}{4}]$ and $q = 1 - p$, then:

$$\frac{\partial F}{\partial x_0} = -12q + 4p \leq 0 \qquad \frac{\partial F}{\partial x_1} = 4p - 4(1-q) = 0$$
$$\frac{\partial F}{\partial x_2} = -3q + p \leq 0 \qquad \frac{\partial F}{\partial x_3} = 11p - 5q(1-1) = 0$$
$$\frac{\partial F}{\partial x_4} = 15p - q - 15p + q = 0 \qquad \frac{\partial F}{\partial x_5} = -p + 3q \geq 0$$
$$\frac{\partial F}{\partial x_6} = -4q + 4q = 0 \qquad \frac{\partial F}{\partial x_7} = -4p + 12q \geq 0$$

Therefore, $A$ is a local optimum for $p \in [\frac{1}{2}, \frac{3}{4}]$. Similarly, it can be shown that $B$ is a local optimum for $p \in [\frac{3}{4}, 1]$ which concludes the proof. □

## 4 Matroid Independence Constraint

Let $\mathcal{M} = (X, \mathcal{I})$ be a matroid. We design an algorithm for the case of submodular maximization subject to a matroid independence constraint, $\max\{f(S) : S \in \mathcal{I}\}$, as follows. The algorithm uses fractional local search to solve the optimization problem $\max\{F(x) : x \in P_t(\mathcal{M})\}$, where $P_t(\mathcal{M}) = P(\mathcal{M}) \cap [0,t]^X$ is a matroid polytope intersected with a box. This technique, which has been used already in [33], is combined with a simulated annealing procedure, where the parameter $t$ is gradually being increased from 0 to 1. (The analogy with simulated annealing is less explicit here; in some sense the system exhibits the most randomness in the middle of the process, when $t = 1/2$.) Finally, the fractional solution is rounded using pipage rounding [4, 33]; we omit this stage from the description of the algorithm.

The main difficulty in designing the algorithm is how to handle the temperature-increasing step. Contrary to the unconstrained problem, we cannot just increment all variables which were previously saturated at $x_i = t$, because this might violate the matroid constraint. Instead, we find a subset of variables that can be increased, by reduction to a bipartite matching problem. We need the following definitions.

DEFINITION 4.1. *Let* 0 *be an extra element not occurring in the ground set* $X$*, and define formally* $\frac{\partial F}{\partial x_0} = 0$. *For* $\mathbf{x} = \frac{1}{N}\sum_{\ell=1}^{n} \mathbf{1}_{I_\ell}$ *and* $i \notin I_\ell$*, we define* $b_\ell(i) = \operatorname{argmin}_{j \in I_\ell \cup \{0\} : I_\ell - j + i \in \mathcal{I}} \frac{\partial F}{\partial x_j}$.

In other words, $b_\ell(i)$ is the least valuable element which can be exchanged for $i$ in the independent set $I_\ell$. Note that such an element must exist due to matroid axioms. We also consider $b_\ell(i) = 0$ as an option in case

$I_\ell + i$ itself is independent. In the following, 0 can be thought of as a special "empty" element, and the partial derivative $\frac{\partial F}{\partial x_0}$ is considered identically equal to zero. By definition, we get the following statement.

LEMMA 4.1. *For $b_\ell(i)$ defined as above, we have $\frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{b_\ell(i)}} = \max_{j \in I_\ell \cup \{0\} : I_\ell - j + i \in \mathcal{I}} \left( \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_j} \right)$.*

The following definition is important for the description of our algorithm.

DEFINITION 4.2. *For $\mathbf{x} = \frac{1}{N} \sum_{\ell=1}^{n} \mathbf{1}_{I_\ell}$, let $A = \{i : x_i = t\}$. We define a bipartite "fractional exchange graph" $G_x$ on $A \cup [N]$ as follows: We have an edge $(i, \ell) \in E$, whenever $i \notin I_\ell$. We define its weight as*

$$w_{i\ell} = \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{b_\ell(i)}} = \max_{j \in I_\ell \cup \{0\} : I_\ell - j + i \in \mathcal{I}} \left( \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_j} \right).$$

We remark that the vertices of the bipartite exchange graph are not elements of $X$ on both sides, but elements on one side and independent sets on the other side.

Algorithm 2 is our complete algorithm for matroid independence constraints. As a subroutine in Step 9, we use the discrete local search algorithm of [22]. The returned solution is a point in $P(\mathcal{M})$; finally, we obtain an integer solution using the pipage rounding technique [5, 33]. We omit this from the description of the algorithm.

THEOREM 4.1. *For any submodular function $f : 2^X \to \mathbb{R}_+$ and matroid $\mathcal{M} = (X, \mathcal{I})$, Algorithm 2 returns with high probability a solution of value at least $0.325 \cdot OPT$ where $OPT = \max_{S \in \mathcal{I}} f(S)$.*

Let us point out some differences between the analysis of this algorithm and the one for unconstrained maximization (Algorithm 1). The basic idea is the same: we obtain certain conditions for partial derivatives at the point of a local optimum. These conditions help us either to conclude that the local optimum already has a good value, or to prove that by relaxing the temperature parameter we gain a certain improvement. We will prove the following lemma which is analogous to Lemma 3.3.

LEMMA 4.2. *Let $\mathbf{x}(t)$ denote the local optimum found by Algorithm 2 at temperature $t < 1 - 1/n$ right after the "Local search" phase, and let $\Phi(t) = F(\mathbf{x}(t))$ denote the value of this local optimum. Also assume that the solution found in "Complementary solution check" phase of the algorithm (Steps 8-10) is always at most $\beta$. Then the function $\Phi(t)$ satisfies*

$$\frac{1-t}{\delta}(\Phi(t+\delta) - \Phi(t)) \geq (1 - 2\delta n^3) OPT - 2\Phi(t) - 2\beta t. \tag{4.10}$$

**Algorithm 2** Simulated Annealing Algorithm for a Matroid Independence Constraint

**Input:** A submodular function $f : 2^X \to \mathbb{R}_+$ and a matroid $\mathcal{M} = (X, \mathcal{I})$.

**Output:** A solution $\mathbf{x} \in P(\mathcal{M})$ such that $F(\mathbf{x}) \geq 0.325 \cdot \max\{f(S) : S \in \mathcal{I}\}$.

1: Let $\mathbf{x} \leftarrow 0$, $N \leftarrow n^4$ and $\delta \leftarrow 1/N$.
2: **Define** $P_t(\mathcal{M}) = P(\mathcal{M}) \cap [0, t]^X$.
3: Maintain a representation of $\mathbf{x} = \frac{1}{N} \sum_{\ell=1}^{N} \mathbf{1}_{I_\ell}$ where $I_\ell \in \mathcal{I}$.
4: **for** $t \leftarrow 0;\ t \leq 1;\ t \leftarrow t + \delta$ **do**
5: **while** there is $\mathbf{v} \in \{\pm \mathbf{e}_i, \mathbf{e}_i - \mathbf{e}_j : i, j \in X\}$ such that $\mathbf{x} + \delta\mathbf{v} \in P_t(\mathcal{M})$ and $F(\mathbf{x} + \delta\mathbf{v}) > F(\mathbf{x})$ **do**
6: $\mathbf{x} := \mathbf{x} + \delta\mathbf{v}$ {**Local search**}
7: **end while**
8: **for** each of the $n + 1$ possible sets $T_{\leq\lambda}(\mathbf{x}) = \{i : x_i \leq \lambda\}$ **do** {**Complementary solution check**}
9: Find a local optimum $B \subseteq T_{\leq\lambda}(\mathbf{x}), B \in \mathcal{I}$ trying to maximize $f(B)$.
10: Remember $\mathbf{1}_B$ for the largest $f(B)$ as a possible candidate for the output of the algorithm.
11: **end for**
12: Form the fractional exchange graph (see Definition 4.2) and find a max-weight matching $M$.
13: Replace $I_\ell$ by $I_\ell - b_\ell(i) + i$ for each edge $(i, \ell) \in M$, and update the point $\mathbf{x} = \frac{1}{N} \sum_{\ell=1}^{N} \mathbf{1}_{I_\ell}$. {**Temperature relaxation**: each coordinate increases by at most $\delta = 1/N$ and hence $\mathbf{x} \in P_{t+\delta}(\mathcal{M})$.}
14: **end for**
15: **return** the best encountered solution $\mathbf{x} \in P(\mathcal{M})$.

We proceed in two steps, again using as an intermediate bound the notion of derivative of $F$ on the line towards the optimum: $G(\mathbf{x}) = (\mathbf{1}_C - \mathbf{x}) \cdot \nabla F(\mathbf{x})$. The plan is to relate the actual gain of the algorithm in the "Temperature relaxation" phase (Steps 12-13) to $G(\mathbf{x})$, and then to argue that $G(\mathbf{x})$ can be compared to the RHS of (4.10). The second part relies on the submodularity of the objective function and is quite similar to the second part of Lemma 3.2 (although slightly more involved).

The heart of the proof is to show that by relaxing the temperature we gain an improvement at least $\frac{\delta}{1-t} G(\mathbf{x})$. As the algorithm suggests, the improvement in this step is related to the weight of the matching obtained in Step 12 of the algorithm. Thus the main goal is to prove that there exists a matching of weight at least $\frac{1}{1-t} G(\mathbf{x})$. We prove this by a combinatorial argument using the local optimality of the current fractional

solution, and an application of König's theorem on edge colorings of bipartite graphs.

Our first goal is to prove Lemma 4.2. As we discussed, the key step is to compare the gain in the temperature relaxation step to the value of the derivative on the line towards the optimum, $G(\mathbf{x}) = (\mathbf{1}_C - \mathbf{x}) \cdot \nabla F(\mathbf{x})$. We prove the following.

LEMMA 4.3. *Let $\mathbf{x}(t)$ be the the local optimum at time $t < 1 - 1/n$. Then*

$$\frac{1-t}{\delta}(F(\mathbf{x}(t+\delta)) - F(\mathbf{x}(t))) \geq G(\mathbf{x}(t)) - n^2\delta \sup\Big|\frac{\partial^2 F}{\partial x_i \partial x_j}\Big|.$$

This lemma can be compared to the first part of the proof of Lemma 3.2, which is not very complicated in the unconstrained case. As we said, the main difficulty here is that relaxing the temperature does not automatically allow us to increase all the coordinates with a positive partial derivative. The reason is that the new fractional solution might not belong to $P_{t+\delta}(\mathcal{M})$. Instead, the algorithm modifies coordinates according to a certain maximum-weight matching found in Step 12. The next lemma shows that the weight of this matching is comparable to $G(\mathbf{x})$.

LEMMA 4.4. *Let $\mathbf{x} = \frac{1}{N}\sum_{\ell=1}^{N} \mathbf{1}_{I_\ell} \in P_t(\mathcal{M})$ be a fractional local optimum, and $C \in \mathcal{I}$ a global optimum. Assume that $(1-t)N \geq n$. Let $G_x$ be the fractional exchange graph defined in Def. 4.2. Then $G_x$ has a matching $M$ of weight*

$$w(M) \geq \frac{1}{1-t} G(\mathbf{x}).$$

*Proof.* We use a basic property of matroids (see [29]) which says that for any two independent sets $C, I \in \mathcal{I}$, there is a mapping $m : C \setminus I \to (I \setminus C) \cup \{0\}$ such that for each $i \in C \setminus I$, $I - m(i) + i$ is independent, and each element of $I \setminus C$ appears at most once as $m(i)$. I.e., $m$ is a matching, except for the special element 0 which can be used as $m(i)$ whenever $I + i \in \mathcal{I}$. Let us fix such a mapping for each pair $C, I_\ell$, and denote the respective mapping by $m_\ell : C \setminus I_\ell \to I_\ell \setminus C$.

Denote by $W$ the sum of all positive edge weights in $G_x$. We estimate $W$ as follows. For each $i \in A \cap C$ and each edge $(i, \ell)$, we have $i \in A \cap C \setminus I_\ell$ and by Lemma 4.1

$$w_{i\ell} = \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{b_\ell(i)}} \geq \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{m_\ell(i)}}.$$

Observe that for $i \in (C \setminus A) \setminus I_\ell$, we get

$$0 \geq \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{m_\ell(i)}}$$

because otherwise we could replace $I_\ell$ by $I_\ell - m_\ell(i) + i$, which would increase the objective function (and for elements outside of $A$, we have $x_i < t$, so $x_i$ can be increased). Let us add up the first inequality over all elements $i \in A \cap C \setminus I_\ell$ and the second inequality over all elements $i \in (C \setminus A) \setminus I_\ell$:

$$\begin{aligned}\sum_{i \in A \cap C \setminus I_\ell} w_{i\ell} &\geq \sum_{i \in C \setminus I_\ell} \left(\frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{m_\ell(i)}}\right) \\ &\geq \sum_{i \in C \setminus I_\ell} \frac{\partial F}{\partial x_i} - \sum_{j \in I_\ell \setminus C} \frac{\partial F}{\partial x_j}\end{aligned}$$

where we used the fact that each element of $I_\ell \setminus C$ appears at most once as $m_\ell(i)$, and $\frac{\partial F}{\partial x_j} \geq 0$ for any element $j \in I_\ell$ (otherwise we could remove it and improve the objective value). Now it remains to add up these inequalities over all $\ell = 1, \ldots, N$:

$$\begin{aligned}\sum_{\ell=1}^{N} \sum_{i \in A \cap C \setminus I_\ell} w_{i\ell} &\geq \sum_{\ell=1}^{N} \left(\sum_{i \in C \setminus I_\ell} \frac{\partial F}{\partial x_i} - \sum_{j \in I_\ell \setminus C} \frac{\partial F}{\partial x_j}\right) \\ &= N \sum_{i \in C} (1 - x_i) \frac{\partial F}{\partial x_i} - N \sum_{j \notin C} x_j \frac{\partial F}{\partial x_j}\end{aligned}$$

using $x_i = \sum_{\ell : i \in I_\ell} \frac{1}{N}$. The left-hand side is a sum of weights over a subset of edges. Hence, the sum of all positive edge weights also satisfies

$$W \geq N \sum_{i \in C} (1 - x_i) \frac{\partial F}{\partial x_i} - N \sum_{j \notin C} x_j \frac{\partial F}{\partial x_j} = N \cdot G(\mathbf{x}).$$

Finally, we apply König's theorem on edge colorings of bipartite graphs: Every bipartite graph of maximum degree $\Delta$ has an edge coloring using at most $\Delta$ colors. The degree of each node $i \in A$ is the number of sets $I_\ell$ not containing $i$, which is $(1-t)N$, and the degree of each node $\ell \in [N]$ is at most the number of elements $n$, by assumption $n \leq (1-t)N$. By König's theorem, there is an edge coloring using $(1-t)N$ colors. Each color class is a matching, and by averaging, the positive edge weights in some color class have total weight

$$w(M) \geq \frac{W}{(1-t)N} \geq \frac{1}{1-t} G(\mathbf{x}).$$

□

The weight of the matching found by the algorithm corresponds to how much we gain by increasing the parameter $t$. Now we can prove Lemma 4.3.

*Proof.* [Lemma 4.3] Assume the algorithm finds a matching $M$. By Lemma 4.4, its weight is

$$w(M) = \sum_{(i,\ell) \in M} \left(\frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{b_\ell(i)}}\right) \geq \frac{1}{1-t} G(\mathbf{x}(t)).$$

If we denote by $\tilde{\mathbf{x}}(t)$ the fractional solution right after the "Temperature relaxation" phase, we have

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + \delta \sum_{(i,\ell)\in M} (\mathbf{e}_i - \mathbf{e}_{b_\ell(i)}).$$

Note that $\mathbf{x}(t+\delta)$ is obtained by applying fractional local search to $\tilde{\mathbf{x}}(t)$. This cannot decrease the value of $F$, and hence

$$F(\mathbf{x}(t+\delta)) - F(\mathbf{x}(t)) \geq F(\tilde{\mathbf{x}}(t)) - F(\mathbf{x}(t))$$
$$= F\left(\mathbf{x}(t) + \delta \sum_{(i,\ell)\in M} (\mathbf{e}_i - \mathbf{e}_{b_\ell(i)})\right) - F(\mathbf{x}(t)).$$

Observe that up to first-order approximation, this increment is given by the partial derivatives evaluated at $\mathbf{x}(t)$. By Lemma A.2, the second-order term is proportional to $\delta^2$:

$$F(\mathbf{x}(t+\delta)) - F(\mathbf{x}(t)) \geq \delta \sum_{(i,\ell)\in M} \left(\frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial x_{b_\ell(i)}}\right) - n^2\delta^2 \sup\left|\frac{\partial^2 F}{\partial x_i \partial x_j}\right|$$

and from above,

$$F(\mathbf{x}(t+\delta)) - F(\mathbf{x}(t)) \geq \frac{\delta}{1-t} G(\mathbf{x}(t)) - n^2\delta^2 \sup\left|\frac{\partial^2 F}{\partial x_i \partial x_j}\right|$$

□

It remains to relate $G(\mathbf{x}(t))$ to the optimum (recall that $OPT = f(C)$), using the complementary solutions found in Step 9. In the next lemma, we show that $G(\mathbf{x})$ is lower bounded by the RHS of equation (4.10).

LEMMA 4.5. *Assume $OPT = f(C)$, $\mathbf{x} \in P_t(\mathcal{M})$, $T_{\leq\lambda}(\mathbf{x}) = \{i : x_i \leq \lambda\}$, and the value of a local optimum on any of the subsets $T_{\leq\lambda}(\mathbf{x})$ is at most $\beta$. Then*

$$G(\mathbf{x}(t)) \geq OPT - 2F(\mathbf{x}) - 2\beta t.$$

*Proof.* Submodularity means that partial derivatives can only decrease when coordinates increase. Therefore by Lemma A.1,

$$\begin{array}{|c|c|}\hline 1 & t \\ \hline 1 & x \\ \hline \end{array} - \begin{array}{|c|c|}\hline t & t \\ \hline x & x \\ \hline \end{array} \leq \sum_{i\in C} (1-x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}}$$

and similarly

$$\begin{array}{|c|c|}\hline t & t \\ \hline x & x \\ \hline \end{array} - \begin{array}{|c|c|}\hline t & 0 \\ \hline x & 0 \\ \hline \end{array} \geq \sum_{j\notin C} x_j \frac{\partial F}{\partial x_j}\Big|_{\mathbf{x}}.$$

Combining these inequalities, we obtain

$$2F(\mathbf{x}(t)) + G(\mathbf{x}(t)) = 2 \begin{array}{|c|c|}\hline t & t \\ \hline x & x \\ \hline \end{array} + \sum_{i\in C} (1-x_i)\frac{\partial F}{\partial x_i} - \sum_{j\notin C} x_j \frac{\partial F}{\partial x_j}$$
$$\geq \begin{array}{|c|c|}\hline 1 & t \\ \hline 1 & x \\ \hline \end{array} + \begin{array}{|c|c|}\hline t & 0 \\ \hline x & 0 \\ \hline \end{array}. \qquad (4.11)$$

Let $A = \{i : x_i = t\}$ (and recall that $x_i \in [0,t]$ for all $i$). By applying the treshold lemma (see Lemma A.3 and the accompanying example with equation (A.2)), we have:

$$\begin{array}{|c|c|}\hline 1 & t \\ \hline 1 & x \\ \hline \end{array} \geq t\ \mathbf{E}\left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline 0 & \\ \hline \end{array}\ \Big|\ \lambda < t\right] + (1-t)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} \qquad (4.12)$$

By another application of Lemma A.3,

$$\begin{array}{|c|c|}\hline t & 0 \\ \hline x & 0 \\ \hline \end{array} \geq t\ \mathbf{E}\left[\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline 0 & \\ \hline \end{array}\ \Big|\ \lambda < t\right] \qquad (4.13)$$

(We discarded the term conditioned on $\lambda \geq t$, where $T_{>\lambda}(\mathbf{x}) = \emptyset$.) It remains to combine this with a suitable set in the complement of $T_{>\lambda}(\mathbf{x})$. Let $S_\kappa$ be a local optimum found inside $T_{\leq\kappa}(\mathbf{x}) = \overline{T_{>\lambda}(\mathbf{x})}$. By Lemma 2.2 in [22], $f(S_\kappa)$ can be compared to any feasible subset of $T_{\leq\kappa}(\mathbf{x})$, e.g. $C_\kappa = C \cap T_{\leq\kappa}(\mathbf{x})$, as follows:

$$\begin{aligned} 2f(S_\kappa) &\geq f(S_\kappa \cup C_\kappa) + f(S_\kappa \cap C_\kappa) \\ &\geq f(S_\kappa \cup C_\kappa) = f(S_\kappa \cup (C \setminus T_{>\kappa}(\mathbf{x}))). \end{aligned}$$

We assume that $f(S_\kappa) \leq \beta$ for any $\kappa$. Let us take expectation over $\lambda \in [0,1]$ uniformly random:

$$\begin{aligned} 2\beta &\geq 2\mathbf{E}\left[f(S_\lambda) \mid \lambda < t\right] \\ &\geq \mathbf{E}\left[f(S_\lambda \cup (C \setminus T_{>\lambda}(\mathbf{x}))) \mid \lambda < t\right]. \end{aligned}$$

Now we can combine this with (4.12) and (4.13):

$$\begin{array}{|c|c|}\hline 1 & t \\ \hline 1 & x \\ \hline \end{array} + \begin{array}{|c|c|}\hline t & 0 \\ \hline x & 0 \\ \hline \end{array} + 2\beta t$$
$$\geq t\mathbf{E}\left[\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline 0 & \\ \hline \end{array} + \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline 0 & \\ \hline \end{array} + f(S_\lambda \cup (C \setminus T_{>\lambda}(\mathbf{x})))\ \Big|\ \lambda < t\right]$$
$$+ (1-t)\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array}$$
$$\geq (1-t)f(C) + t\left[\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline 0 & \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline 1 & \\ \hline \end{array}\right]$$
$$\geq (1-t)f(C) + tf(C) = f(C) = OPT.$$

where the last two inequalities follow from submodularity. Together with (4.11), this finishes the proof. □

*Proof.* [Lemma 4.2] By Lemma 4.3 and 4.5, we get

$$\frac{1-t}{\delta}(\Phi(t+\delta)-\Phi(t)) = \frac{1-t}{\delta}(F(\mathbf{x}(t+\delta))-F(\mathbf{x}(t))) \geq OPT - 2F(\mathbf{x}) - 2\beta t - n^2\delta \sup\Big|\frac{\partial^2 F}{\partial x_i \partial x_j}\Big|.$$

We have $|\frac{\partial^2 F}{\partial x_i \partial x_j}| \leq 2\max|f(S)| \leq 2nOPT$, which implies the lemma. □

Now by taking $\delta \to 0$, the statement of Lemma 4.2 leads naturally to the following differential equation:

$$(1-t)\Phi'(t) \geq OPT - 2\Phi(t) - 2t\beta.$$

This differential equation is very similar to the one we obtained in Section 3 can be solved analytically as well.

We start from initial conditions corresponding to the 0.309-approximation of [33], which implies that a fractional local optimum at $t_0 = \frac{1}{2}(3-\sqrt{5})$ has value $v_0 \geq \frac{1}{2}(1-t_0) \simeq 0.309$. We prove that there is a value $\beta > 0.325$ such that for some value of $t$ (which turns out to be roughly 0.53), we get $\Phi(t) \geq \beta$.

Let us assume that $OPT = 1$. Starting from an initial point $F(t_0) = v_0$, the solution turns out to be

$$\Phi(t) \geq \frac{1}{2} + \beta - 2\beta t - \frac{(1-t)^2}{(1-t_0)^2}\left(\frac{1}{2} + \beta - 2\beta t_0 - v_0\right).$$

We start from initial conditions corresponding to the 0.309-approximation of [33]. It is proved in [33] that a fractional local optimum at $t_0 = (1-t_0)^2 = \frac{1}{2}(3-\sqrt{5})$ has value $v_0 \geq \frac{1}{2}(1-t_0) \simeq 0.309$. Therefore, we obtain the following solution for $t \geq \frac{1}{2}(3-\sqrt{5})$:

$$\Phi(t) \geq \frac{1}{2} + \beta - 2\beta t - (1-t)^2\left(\frac{1}{2} - 2\beta + \frac{2\beta}{3-\sqrt{5}}\right).$$

We solve for $\beta$ such that the maximum of the right-hand side equals $\beta$. The solution is

$$\beta = \frac{1}{8}\left(2+\sqrt{5}\right)\left(-5+\sqrt{5}+\sqrt{-2+6\sqrt{5}}\right).$$

Then, for some value of $t$ (which turns out to be roughly 0.53), we have $\Phi(t) \geq \beta$. It can be verified that $\beta > 0.325$; this proves Theorem 4.1.

## 5 Hardness of approximation

In this section, we improve the hardness of approximating several submodular maximization problems subject to additional constraints (i.e. $\max\{f(S) : S \in \mathcal{F}\}$), assuming the value oracle model. We use the method of *symmetry gap* [33] to derive these new results. This method can be summarized as follows. We start with a fixed instance $\max\{f(S) : S \in \mathcal{F}\}$ which is symmetric under a certain group of permutations of the ground set $X$. We consider the multilinear relaxation of this instance, $\max\{F(\mathbf{x}) : \mathbf{x} \in P(\mathcal{F})\}$. We compute the *symmetry gap* $\gamma = \overline{OPT}/OPT$, where $OPT = \max\{F(\mathbf{x}) : \mathbf{x} \in P(\mathcal{F})\}$ is the optimum of the relaxed problem and $\overline{OPT} = \max\{F(\bar{\mathbf{x}}) : \mathbf{x} \in P(\mathcal{F})\}$ is the optimum over all *symmetric* fractional solutions, i.e. satisfying $\sigma(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$ for any $\sigma \in \mathcal{G}$. Due to [33, Theorem 1.6], we obtain hardness of $(1+\epsilon)\gamma$-approximation for a class of related instances, as follows.

THEOREM 5.1. ([33]) *Let $\max\{f(S) : S \in \mathcal{F}\}$ be an instance of a nonnegative submodular maximization problem with symmetry gap $\gamma = \overline{OPT}/OPT$. Let $\mathcal{C}$ be the class of instances $\max\{\tilde{f}(S) : S \in \tilde{\mathcal{F}}\}$ where $\tilde{f}$ is nonnegative submodular and $\tilde{\mathcal{F}}$ is a "refinement" of $\mathcal{F}$. Then for every $\epsilon > 0$, any $(1+\epsilon)\gamma$-approximation algorithm for the class of instances $\mathcal{C}$ would require exponentially many value queries to $\tilde{f}(S)$.*

For a formal definition of "refinement", we refer to [33, Definition 1.5]. Intuitively, these are "blown-up" copies of the original family of feasible sets, such that the constraint is of the same type as the original instance (e.g. cardinality, matroid independence and matroid base constraints are preserved).

**Directed hypergraph cuts.** Our main tool in deriving the new results is a construction using a variant of the Max Di-cut problem in *directed hypergraphs*. We consider the following variant of directed hypergraphs.

DEFINITION 5.1. *A directed hypergraph is a pair $H = (X, E)$, where $E$ is a set of directed hyperedges $(U, v)$, where $U \subset X$ is a non-empty subset of vertices and $v \notin U$ is a vertex in $X$.*

*For a set $S \subset X$, we say that a hyperedge $(U, v)$ is cut by $S$, or $(U, v) \in \delta(S)$, if $U \cap S \neq \emptyset$ and $v \notin S$.*

Note that a directed hyperedge should have exactly one head. An example of a directed hypergraph is shown in Figure 3. We will construct our hard examples as Max Di-cut instances on directed hypergraphs. It is easy to see that the number (or weight) of hyperedges cut by a set $S$ is indeed submodular as a function of $S$. Other types of directed hypergraphs have been considered, in particular with hyperedges of multiple heads and tails, but a natural extension of the cut function to such hypergraphs is no longer submodular.

In the rest of this section, first we present our hardness result for maximizing submodular functions subject to a matroid base constraint (when the base packing number of the matroid is at least 2). Then, in subsections 5.1 we prove a stronger hardness result when the base packing number of the matroid is smaller than 2. Finally in subsections 5.2 and 5.3 we prove new hardness results for maximizing submodular functions subject to a matroid independence or a cardinality constraint.

THEOREM 5.2. *There exist instances of the problem* $\max\{f(S) : S \in \mathcal{B}\}$*, where* $f$ *is a nonnegative submodular function,* $\mathcal{B}$ *is a collection of matroid bases of packing number at least* 2*, and any* $(1-e^{-1/2}+\epsilon)$*-approximation for this problem would require exponentially many value queries for any* $\epsilon > 0$*.*

We remark that $1 - e^{-1/2} < 0.394$, and only hardness of $(0.5 + \epsilon)$-approximation was previously known in this setting.

**Instance 1.** Consider the directed hypergraph in Figure 3, with the set of vertices $X = A \cup B$ and two hyperedges $(\{a_1, \ldots, a_k\}, a)$ and $(\{b_1, \ldots, b_k\}, b)$. Let $f$ be the cut function on this hypergraph, and let $\mathcal{M}_{A,B}$ be a partition matroid whose independent sets contain at most one vertex from each of the sets $A$ and $B$. Let $\mathcal{B}_{A,B}$ be the bases of $\mathcal{M}_{A,B}$ (i.e. $\mathcal{B}_{A,B} = \{S : |S \cap A| = 1 \ \& \ |S \cap B| = 1\}$). Note that there exist two disjoint bases in this matroid and the base packing number of $\mathcal{M}$ is equal to 2. An optimum solution is for example $S = \{a, b_1\}$ with $OPT = 1$.

In order to apply Theorem 5.1 we need to compute the symmetry gap of this instance $\gamma = \overline{OPT}/OPT$. We remark in the blown-up instances, $\overline{OPT}$ corresponds to the maximum value that any algorithm can obtain, while $OPT = 1$ is the actual optimum. The definition of $\overline{OPT}$ depends on the symmetries of our instance, which we describe in the following lemma.

LEMMA 5.1. *There exists a group* $\mathcal{G}$ *of permutations such that Instance 1 is symmetric under* $\mathcal{G}$*, in the sense that* $\forall \sigma \in \mathcal{G}$*;*

$$f(S) = f(\sigma(S)), \qquad S \in \mathcal{B}_{A,B} \Leftrightarrow \sigma(S) \in \mathcal{B}_{A,B}. \quad (5.14)$$

*Moreover, for any two vertices* $i, j \in A$ *(or* $B$*), the probability that* $\sigma(i) = j$ *for a uniformly random* $\sigma \in \mathcal{G}$ *is equal to* $1/|A|$ *(or* $1/|B|$ *respectively).*

*Proof.* Let $\Pi$ be the set of the following two basic permutations

$$\Pi = \begin{cases} \sigma_1 : \sigma_1(a) = b, \sigma_1(b) = a, \sigma_1(a_i) = b_i, \sigma_1(b_i) = a_i \\ \sigma_2 : \sigma_2(a) = a, \sigma_2(b) = b, \sigma_2(a_i) = a_{i+1}, \sigma_2(b_i) = b_i \end{cases}$$

where $\sigma_1$ swaps the vertices of the two hyperedges and $\sigma_2$ only rotates the tail vertices of one of the hyperedges. (Indices are taken modulo $k$.) It is easy to see that both of these permutations satisfy equation (5.14). Therefore, our instance is *invariant* under each of the basic permutations and also under any permutation generated by them. Now let $\mathcal{G}$ be the set of all the permutations that are generated by $\Pi$. $\mathcal{G}$ is a *group* and under this group of symmetries all the elements in $A$ (and $B$) are equivalent. In other words, for any three vertices $i, j, k \in A$ (or $B$), the number of permutations $\sigma \in G$ such that $\sigma(i) = j$ is equal to the number of permutations such that $\sigma(i) = k$. □

Using the above lemma we may compute the *symmetrization* of a vector $\mathbf{x} \in [0,1]^X$ which will be useful in computing $\overline{OPT}$ [33]. For any vector $\mathbf{x} \in [0,1]^X$, the "symmetrization of $\mathbf{x}$" is:

$$\bar{\mathbf{x}} = \mathbf{E}_{\sigma \in \mathcal{G}}\left[\sigma(\mathbf{x})\right] = \begin{cases} \bar{x}_a = \bar{x}_b = \frac{1}{2}(x_a + x_b) \\ \bar{x}_{a_j} = \bar{x}_{b_j} = \frac{1}{2k}\sum_{i=1}^{k}(x_{a_i} + x_{b_i}) \end{cases} \quad (5.15)$$

where $\sigma(\mathbf{x})$ denotes $\mathbf{x}$ with coordinates permuted by $\sigma$. Now we are ready to prove Theorem 5.2.

*Proof.* [Theorem 5.2] We need to compute the value of symmetry gap $\gamma = \overline{OPT} = \max\{F(\bar{\mathbf{x}}) : \mathbf{x} \in P(\mathcal{B}_{A,B})\}$, where $F$ is the multilinear relaxation of $f$ and $P(\mathcal{B}_{A,B})$ is the convex hull of the bases in $\mathcal{B}_{A,B}$. For any vector $\mathbf{x} \in [0,1]^X$, we have

$$\mathbf{x} \in P(\mathcal{B}_{A,B}) \Leftrightarrow \begin{cases} x_a + x_b = 1 \\ \sum_{i=1}^{k}(x_{a_i} + x_{b_i}) = 1. \end{cases} \quad (5.16)$$

By equation (5.15) we know that the vertices in each of the sets $A$, $B$ have the same value in $\bar{\mathbf{x}}$. Using equation (5.16), we obtain $\bar{x}_a = \bar{x}_b = \frac{1}{2}$ and $\bar{x}_{a_i} = \bar{x}_{b_i} = \frac{1}{2k}$ for all $1 \le i \le k$, which yields a unique symmetrized solution $\bar{\mathbf{x}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2k}, \ldots, \frac{1}{2k})$.

Now we can simply compute $\overline{OPT} = F(\frac{1}{2}, \frac{1}{2}, \frac{1}{2k}, \ldots, \frac{1}{2k})$. Note that by definition a hyperedge will be cut by a random set $S$ if and only if at least one of its tails is included in $S$ while its head is not included. Therefore

$$\begin{aligned} \overline{OPT} &= F\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2k}, \ldots, \frac{1}{2k}\right) \\ &= 2\left[\frac{1}{2}\left(1 - \left(1 - \frac{1}{2k}\right)^k\right)\right] \simeq 1 - e^{-\frac{1}{2}}, \end{aligned}$$

for sufficiently large $k$. By applying Theorem 5.1, it can be seen that the refined instances are instances of

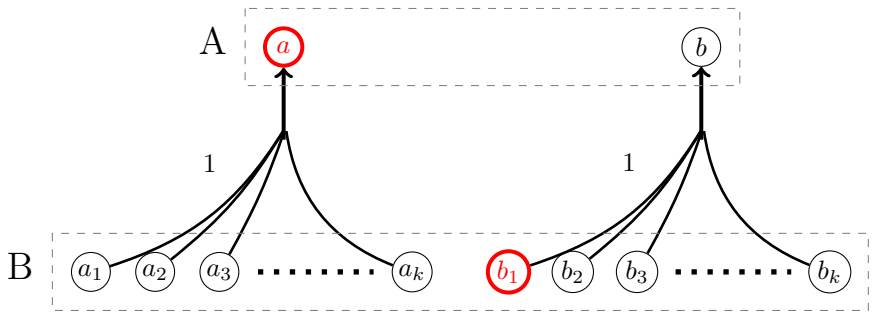


Figure 3: Example for maximizing a submodular function subject to a matroid base constraint; the objective function is a directed hypergraph cut function, and the constraint is that we should pick exactly 1 element of $A$ and 1 element of $B$.

submodular maximization over the bases of a matroid where the ground set is partitioned into $A \cup B$ and we have to take half of the elements of $A$ and $\frac{1}{2k}$ fraction of the elements in $B$. Thus the base packing number of the matroid in the refined instances is also 2 which implies the theorem. □

**5.1 General matroid base constraints.** It is shown in [33] that it is hard to approximate submodular maximization subject to a matroid base constraint with fractional base packing number $\nu = \ell/(\ell-1)$, $\ell \in \mathbb{Z}$, better than $1/\ell$. We showed in Theorem 5.2 that for $\ell = 2$, the threshold of $1/2$ can be improved to $1-e^{-1/2}$. More generally, we show the following.

THEOREM 5.3. *There exist instances of the problem* $\max\{f(S) : S \in \mathcal{B}\}$, *such that a* $(1-e^{-1/\ell}+\epsilon)$ *approximation for any* $\epsilon > 0$ *would require exponentially many value queries. Here* $f(S)$ *is a nonnegative submodular function, and* $\mathcal{B}$ *is a collection of bases in a matroid with fractional base packing number* $\nu = \ell/(\ell-1), \ell \in \mathbb{Z}$.

*Proof.* Let $\nu = \frac{\ell}{\ell-1}$. Consider the hypergraph $H$ in Figure 3, with $\ell$ instead of 2 hyperedges. Similarly let $A$ ($B$) be the set of head (tail) vertices respectively, and let the feasible sets be those that contain $\ell - 1$ vertices of $A$ and one vertex of $B$. (i.e. $\mathcal{B} = \{S : |S \cap A| = \ell - 1 \ \& \ |S \cap B| = 1)$. The optimum can simply select the heads of the first $\ell - 1$ hyperedges and one of the tails of the last one, thus the value of $OPT = 1$ remains unchanged. On the other hand, $\overline{OPT}$ will decrease since the number of symmetric elements has increased and there is a greater chance to miss a hyperedge. Similar to the proof of Lemma 5.1 and Theorem 5.2 we obtain a unique symmetrized vector $\bar{\mathbf{x}} = (\frac{1}{\ell}, \frac{1}{\ell}, \ldots, \frac{1}{\ell}, \frac{1}{k\ell}, \frac{1}{k\ell}, \ldots, \frac{1}{k\ell})$. Therefore,

$$\gamma = \overline{OPT} = F(\bar{\mathbf{x}}) = \ell \left[\frac{1}{\ell}\left(1 - \left(1 - \frac{1}{k\ell}\right)^k\right)\right] \simeq 1 - e^{-1/\ell},$$

for sufficiently large $k$. Also it is easy to see that the feasible sets of the refined instances, which are indeed the bases of a matroid, are those that contain a $\frac{\ell-1}{\ell}$ fraction of the vertices in $A$ and a $\frac{1}{k\ell}$ fraction of vertices in $B$. Therefore the fractional base packing number of the refined instances is equal to $\frac{\ell}{\ell-1}$. □

**5.2 Matroid independence constraint.** In this subsection we focus on the problem of maximizing a submodular function subject to a matroid independence constraint. Similarly to Section 5.1, we construct our hard instances using directed hypergraphs.

THEOREM 5.4. *There exist instances of the problem* $\max\{f(S) : S \in \mathcal{I}\}$ *where* $f$ *is nonnegative submodular and* $\mathcal{I}$ *are independent sets in a matroid such that a* 0.478*-approximation would require exponentially many value queries.*

It is worth noting that the example we considered in Theorem 5.2 does not imply any hardness factor better than $1/2$ for the matroid independence problem. The reason is that for the vector $\bar{\mathbf{x}} = (0, 0, \frac{1}{2k}, \ldots, \frac{1}{2k})$, which is contained in the matroid polytope $P(\mathcal{M})$, the value of the multilinear relaxation is $1/2$. In other words, it is better for an algorithm not to select any vertex in the set $A$, and try to select as much as possible from $B$.

**Instance 2.** To resolve this issue, we perturb the instance by adding an undirected edge $(a, b)$ of weight $1 - \alpha$ and we decrease the weight of the hyperedges to $\alpha$, where the value of $\alpha$ will be optimized later (see Figure 4). The objective function is again the (directed) cut function, where the edge $(a, b)$ contributes $1 - \alpha$ if we pick exactly one of vertices $a, b$. Therefore the value of the optimum remains unchanged, $OPT = \alpha + (1 - \alpha) = 1$. On the other hand the optimal symmetrized vector $\bar{\mathbf{x}}$ should have a non-zero value for the head vertices, otherwise

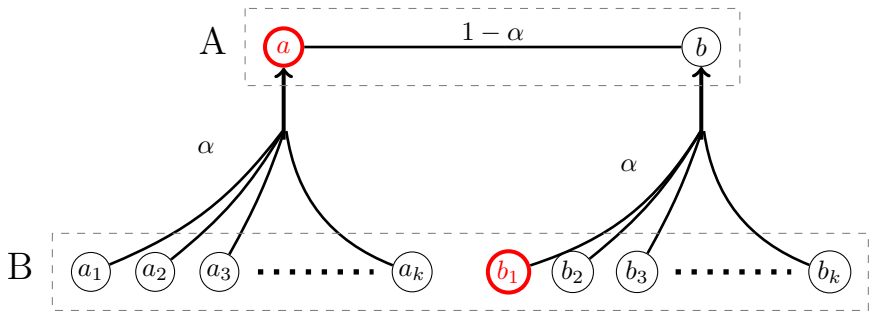


Figure 4: Example for maximizing a submodular function subject to a matroid independence constraint; the hypergraph contains two directed hyperedges of weight $\alpha$ and the edge $(a, b)$ of weight $1 - \alpha$; the constraint is that we pick at most one vertex from each of $A$ and $B$.

the edge $(a, b)$ would not have any contribution to $F(\bar{\mathbf{x}})$.

*Proof.* [Theorem 5.4] Let $H$ be the hypergraph of Figure 4, and consider the problem $\max\{f(S) : S \in \mathcal{I}\}$, where $f$ is the cut function of $H$ and $\mathcal{I}$ is the set of independent sets of the matroid $\mathcal{M}_{A,B}$ defined in subsection 5.2. Observe that Lemma 5.1 can be applied to our instance as well, thus we may use equation (5.15) to obtain the symmetrized vectors $\bar{\mathbf{x}}$. Moreover, the matroid polytope can be described by the following equations:

$$x \in P(\mathcal{M}_{A,B}) \Leftrightarrow \begin{cases} x_a + x_b \leq 1 \\ \sum_{i=1}^{k} (x_{a_i} + x_{b_i}) \leq 1. \end{cases} \tag{5.17}$$

Since the vertices of the set $B$ only contribute as tails of hyperedges, the value of $F(\bar{\mathbf{x}})$ can only increase if we increase the value of $\bar{\mathbf{x}}$ on the vertices in $B$. Therefore, we can assume (using equations (5.15) and (5.17)) that

$$\bar{x}_a = \bar{x}_b \leq \tfrac{1}{2}$$
$$\bar{x}_{a_1} = \bar{x}_{b_1} = \ldots = \bar{x}_{a_k} = \bar{x}_{b_k} = \tfrac{1}{2k}.$$

Let $\bar{x}_a = q$; we may compute the value of $\overline{OPT}$ as follows:

$$\overline{OPT} = F(\bar{\mathbf{x}}) = 2\alpha \left[ (1-q) \left( 1 - (1 - \frac{1}{k})^k \right) \right] + (1-\alpha) \left[ 2q(1-q) \right],$$

where $q \leq 1/2$. By optimizing numerically over $\alpha$, we find that the smallest value of $\overline{OPT}$ is obtained when $\alpha \simeq 0.3513$. In this case we have $\gamma = \overline{OPT} \simeq 0.4773$. Also, similarly to Theorem 5.2, the refined instances are in fact instances of a submodular maximization problem over independent sets of a matroid (a partition matroid whose ground set is partitioned into $A \cup B$ and we have to take at most half of the elements of $A$ and $1/2k$ fraction of elements in $B$). □

**5.3 Cardinality constraint.** Although we do not know how to prove the hardness of maximizing general submodular functions without any additional constraint to a factor smaller than $1/2$, we can show that adding a simple cardinality constraint makes a $1/2$-approximation impossible. In particular, we show that it is hard to approximate a submodular function subject to a cardinality constraint within a factor of 0.491.

COROLLARY 5.1. *There exist instances of the problem* $\max\{f(S) : |S| \leq \ell\}$ *with* $f$ *nonnegative submodular such that a* 0.491*-approximation would require exponentially many value queries.*

We remark that a related problem, $\max\{f(S) : |S| = k\}$, is at least as difficult to approximate: we can reduce $\max\{f(S) : |S| \leq \ell\}$ to it by trying all possible values $k = 0, 1, 2, \ldots, \ell$.

*Proof.* Let $\ell = 2$, and let $H$ be the hypergraph we considered in previous theorem and $f$ be the cut function of $H$. Similar to the proof of Theorem 5.4, we have $OPT = 1$ and we may use equation (5.15) to obtain the value of $\bar{\mathbf{x}}$. In this case the feasibility polytope will be

$$x \in P(|S| \leq 2) \Leftrightarrow x_a + x_b + \sum_{i=1}^{k} (x_{a_i} + x_{b_i}) \leq 2, \tag{5.18}$$

however, we may assume that we have equality for the maximum value of $F(\bar{\mathbf{x}})$, otherwise we can simply increase the $\bar{\mathbf{x}}$ value of the tail vertices in $B$ and this can only increase $F(\bar{\mathbf{x}})$. Let $\bar{x}_a = q$ and $x_{a_1} = p$ and $z = kp$. Using equations (5.15) and (5.18) we have

$$2q + 2kp = 2 \Rightarrow kp = z = 1 - q.$$

Finally, we can compute the value of $\overline{OPT}$:

$$\begin{aligned} \overline{OPT} = F(\bar{\mathbf{x}}) &= 2\alpha \left[ (1-q) \left( 1 - (1-p)^k \right) \right] \\ &\quad + (1-\alpha) \left[ 2q(1-q) \right] \\ &= 2\alpha z (1 - e^{-z}) + 2(1-\alpha) z (1-z). \end{aligned}$$

Again by optimizing over $\alpha$, the smallest value of $\overline{OPT}$ is obtained when $\alpha \simeq 0.15$. In this case we have $\gamma \simeq 0.49098$. The refined instances are instances of submodular maximization subject to a cardinality constraint, where the constraint is to choose at most $\frac{1}{k+1}$ fraction of the all the elements in the ground set. □

**Acknowledgment.** We would like to thank Tim Roughgarden for stimulating discussions.

## References


[1] P. Austrin. Improved inapproximability for submodular maximization, *Proc. of APPROX 2010*, 12–24.

[2] D. Bertsimas and J. Tsitsiklis. Simulated annealing, *Statistical Science* 8:1 (1993), 10–15.

[3] I. Bezáková, D. Štefankovič, V. Vazirani and E. Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM Journal of Computing* 37:5 (2008), 1429–1454.

[4] G. Calinescu, C. Chekuri, M. Pál and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, *Proc. of 12th IPCO* (2007), 182–196.

[5] G. Calinescu, C. Chekuri, M. Pál and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint, to appear in *SIAM J. on Computing*.

[6] C. Chekuri, J. Vondrák and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures, *Proc. of* $51^{th}$ *IEEE FOCS* (2010).

[7] U. Feige. A threshold of $\ln n$ for approximating Set Cover, *Journal of the ACM* 45 (1998), 634–652.

[8] U. Feige and M. X. Goemans. Approximating the value of two-prover systems, with applications to MAX-2SAT and MAX-DICUT, *Proc. of the 3rd Israel Symposium on Theory and Computing Systems*, Tel Aviv (1995), 182–189.

[9] U. Feige, V. Mirrokni and J. Vondrák. Maximizing non-monotone submodular functions, *Proc. of* $48^{th}$ *IEEE FOCS* (2007), 461–471.

[10] M. L. Fisher, G. L. Nemhauser and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions II, *Mathematical Programming Study* 8 (1978), 73–87.

[11] L. Fleischer, S. Fujishige and S. Iwata. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions, *Journal of the ACM* 48:4 (2001), 761–777.

[12] G. Goel, C. Karande, P. Tripathi and L. Wang. Approximability of combinatorial problems with multi-agent submodular cost functions, *Proc. of* $50^{th}$ *IEEE FOCS* (2009), 755–764.

[13] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of the ACM* 42 (1995), 1115–1145.

[14] B. Goldengorin, G. Sierksma, G. Tijsssen and M. Tso. The data correcting algorithm for the minimization of supermodular functions, *Management Science*, 45:11 (1999), 1539–1551.

[15] B. Goldengorin, G. Tijsssen and M. Tso. The maximization of submodular functions: Old and new proofs for the correctness of the dichotomy algorithm, *SOM Report*, University of Groningen (1999).

[16] M. Grötschel, L. Lovász and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1:2 (1981), 169–197.

[17] A. Gupta, A. Roth, G. Schoenebeck and K. Talwar. Constrained non-monotone submodular maximization: offline and secretary algorithms, manuscript, 2010.

[18] S. Iwata and K. Nagano. Submodular function minimization under covering constraints, In *Proc. of* $50^{th}$ *IEEE FOCS* (2009), 671–680.

[19] A. T. Kalai and S Vempala. Simulated annealing for convex optimization, *Math. of Operations Research*, 31:2 (2006), 253–266.

[20] V. R. Khachaturov. Mathematical methods of regional programming (in Russian), *Nauka, Moscow*, 1989.

[21] A. Kulik, H. Shachnai and T. Tamir. Maximizing submodular functions subject to multiple linear constraints, *Proc. of* $20^{th}$ *ACM-SIAM SODA* (2009), 545–554.

[22] J. Lee, V. Mirrokni, V. Nagarajan and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints, *Proc. of* $41^{st}$ *ACM STOC* (2009), 323-332.

[23] J. Lee, M. Sviridenko and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties, *Proc. of APPROX 2009*, 244–257.

[24] H. Lee, G. Nemhauser and Y. Wang. Maximizing a submodular function by integer programming: Polyhedral results for the quadratic case, *European Journal of Operational Research* 94 (1996), 154–166.

[25] L. Lovász. Submodular functions and convexity. A. Bachem et al., editors, *Mathematical Programmming: The State of the Art*, 1983, 235–257.

[26] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm, In *Proc.* $44^{th}$ *IEEE FOCS* (2003), 650–659.

[27] T. Robertazzi and S. Schwartz. An accelated sequential algorithm for producing D-optimal designs, *SIAM Journal on Scientific and Statistical Computing* 10 (1989), 341–359.

[28] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory, Series B* 80 (2000), 346–355.

[29] A. Schrijver. *Combinatorial optimization - polyhedra and efficiency.* Springer, 2003.

[30] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds, *Proc. of 49th IEEE FOCS* (2008), 697–706.

[31] D. Štefankovič, S. Vempala and E. Vigoda. Adaptive simulated annealing: a near-optimal connection be-

tween sampling and counting, *Journal of the ACM* 56:3 (2009), 1–36.

[32] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model, *Proc. of* $40^{th}$ *ACM STOC* (2008), 67–74.

[33] J. Vondrák. Symmetry and approximability of submodular maximization problems, *Proc. of* $50^{th}$ *IEEE FOCS* (2009), 651–670.

## A Miscellaneous Lemmas

Let $F(\mathbf{x})$ be the multilinear extension of a submodular function. The first lemma says that if we increase coordinates from $\mathbf{x}$ to $\mathbf{x}' \geq \mathbf{x}$, then the increase in $F(\mathbf{x})$ is *at most* that given by partial derivatives at $\mathbf{x}$, and *at least* that given by partial derivatives at $\mathbf{x}'$.

LEMMA A.1. *If $F : [0,1]^X \to \mathbb{R}$ is the multilinear extension of a submodular function, and $\mathbf{x}' \geq \mathbf{x}$ where $\mathbf{y} \geq 0$, then*

$$F(\mathbf{x}') \leq F(\mathbf{x}) + \sum_{i \in X} (x'_i - x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}}.$$

*Similarly,*

$$F(\mathbf{x}') \geq F(\mathbf{x}) + \sum_{i \in X} (x'_i - x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}'}.$$

*Proof.* Since $F$ is the multilinear extension of a submodular function, we know that $\frac{\partial^2 F}{\partial x_i \partial x_j} \leq 0$ for all $i, j$ [4]. This means that whenever $\mathbf{x} \leq \mathbf{x}'$, the partial derivatives at $\mathbf{x}'$ cannot be larger than at $\mathbf{x}$:

$$\frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} \geq \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}'}.$$

Therefore, between $\mathbf{x}$ and $\mathbf{x}'$, the highest partial derivatives are attained at $\mathbf{x}$, and the lowest at $\mathbf{x}'$. By integrating along the line segment between $\mathbf{x}$ and $\mathbf{x}'$, we obtain

$$\begin{aligned} F(\mathbf{x}') - F(\mathbf{x}) &= \int_0^1 (\mathbf{x}' - \mathbf{x}) \cdot \nabla F(\mathbf{x} + t(\mathbf{x}' - \mathbf{x})) dt \\ &= \sum_{i \in X} \int_0^1 (x'_i - x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x} + t(\mathbf{x}' - \mathbf{x})} dt. \end{aligned}$$

If we evaluate the partial derivatives at $\mathbf{x}$ instead, we get

$$F(\mathbf{x}') - F(\mathbf{x}) \leq \sum_{i \in X} (x'_i - x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}}.$$

If we evaluate the partial derivatives at $\mathbf{x}'$, we get

$$F(\mathbf{x}') - F(\mathbf{x}) \geq \sum_{i \in X} (x'_i - x_i) \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}'}.$$

□

For a small increase in each coordinate, the partial derivatives give a good approximation of the change in $F$; this is a standard analytic argument, which we formalize in the next lemma.

LEMMA A.2. *Let $F : [0,1]^X \to \mathbb{R}$ be twice differentiable, $\mathbf{x} \in [0,1]^X$ and $\mathbf{y} \in [-\delta, \delta]^X$. Then*

$$\Big| F(\mathbf{x} + \mathbf{y}) - F(\mathbf{x}) - \sum_{i \in X} y_i \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} \Big| \leq \delta^2 n^2 \sup \Big| \frac{\partial^2 F}{\partial x_i \partial x_j} \Big|,$$

*where the supremum is taken over all $i, j$ and all points in $[0,1]^X$.*

*Proof.* Let $M = \sup |\frac{\partial^2 F}{\partial x_i \partial x_j}|$. Since $F$ is twice differentiable, any partial derivative can change by at most $\delta M$ when a coordinate changes by at most $\delta$. Hence,

$$-\delta n M \leq \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x} + t\mathbf{y}} - \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} \leq \delta n M$$

for any $t \in [0,1]$. By the fundamental theorem of calculus,

$$\begin{aligned} F(\mathbf{x} + \mathbf{y}) &= F(\mathbf{x}) + \sum_{i \in X} \int_0^1 y_i \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x} + t\mathbf{y}} dt \\ &\leq F(\mathbf{x}) + \sum_{i \in X} y_i \left( \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} + \delta n M \right) \\ &\leq F(\mathbf{x}) + \sum_{i \in X} y_i \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} + \delta^2 n^2 M. \end{aligned}$$

Similarly we get

$$F(\mathbf{x} + \mathbf{y}) \geq F(\mathbf{x}) + \sum_{i \in X} y_i \frac{\partial F}{\partial x_i}\Big|_{\mathbf{x}} - \delta^2 n^2 M.$$

□

The following "threshold lemma" appears as Lemma A.4 in [33]. We remark that the expression $\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))\right]$ defined below is an alternative definition of the Lovász extension of $f$.

LEMMA A.3. (THRESHOLD LEMMA) *For $\mathbf{y} \in [0,1]^X$ and $\lambda \in [0,1]$, define $T_{>\lambda}(\mathbf{y}) = \{i : y_i > \lambda\}$. If $F$ is the multilinear extension of a submodular function $f$, then for $\lambda \in [0,1]$ uniformly random*

$$F(\mathbf{y}) \geq \mathbf{E}\left[f(T_{>\lambda}(\mathbf{y}))\right].$$

Since we apply this lemma in various places of the paper let us describe some applications of it in detail.

*Example.* In this example we apply the threshold lemma to the vector $\mathbf{x} = p\mathbf{1}_{A\cap C} + (1-p)\mathbf{1}_{B\cap C}$. Here $C$ represents the optimum set, $B = \overline{A}$ and $1/2 < p < 1$. If $\lambda \in [0,1]$ is chosen uniformly at random we know $0 < \lambda \leq 1-p$ with probability $1-p$, $1-p < \lambda \leq p$ with probability $2p-1$ and $p < \lambda \leq 1$ with probability $1-p$. Therefore by Lemma A.3 we have:

$$\begin{aligned} F(\mathbf{x}) &\geq (1-p)\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))|\lambda \leq 1-p\right] \\ &\quad + (2p-1)\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))|1-p < \lambda \leq p\right] \\ &\quad + (1-p)\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))|p < \lambda \leq 1\right] \\ &= (1-p)\mathbf{E}\left[f(C)\right] + (2p-1)\mathbf{E}\left[f(A\cap C)\right] \\ &\quad + (1-p)\mathbf{E}\left[f(\emptyset)\right] \end{aligned}$$

or equivalently we can write

$$\begin{array}{|c|c|}\hline \text{p} & 0 \\ \hline \text{1-p} & 0 \\ \hline \end{array} \geq (1-p) \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} + (2p-1) \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array} + (1-p) \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array}. \tag{A.1}$$

In the next example we consider a more complicated application of the threshold lemma.

*Example.* Consider the vector $\mathbf{x}$ where $x_i = 1$ for $i \in C$, $x_i = t$ for $i \in A \setminus C$ and $x_i < t$ for $i \in B \setminus C$. In this case, we denote

$$F(\mathbf{x}) = \begin{array}{|c|c|}\hline 1 & \text{t} \\ \hline 1 & \text{x} \\ \hline \end{array}.$$

Again $C$ is the optimal set and $B = \overline{A}$. In this case if we apply the threshold lemma, we get a random set which can contain a part of the block $B \setminus C$. In particular, observe that if $\lambda \leq t$, then $T_{>\lambda}(\mathbf{x})$ contains all the elements in $\overline{B \setminus C}$, and depending on the value of $\lambda$, elements in $B \setminus C$ that are greater than $\lambda$. We denote the value of such a set by

$$f(T_{>\lambda}(\mathbf{x})) = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \cline{2-2} & 0 \\ \hline \end{array}$$

where the right-hand lower block is divided into two parts depending on the threshold $\lambda$. Therefore

$$F(\mathbf{x}) \geq t\,\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))|\lambda \leq t\right] + (1-t)\mathbf{E}\left[f(T_{>\lambda}(\mathbf{x}))|\lambda > t\right],$$

can be written equivalently as

$$\begin{array}{|c|c|}\hline 1 & \text{t} \\ \hline 1 & \text{x} \\ \hline \end{array} \geq t\;\mathbf{E}\left[\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \cline{2-2} & 0 \\ \hline \end{array}\;\middle|\;\lambda \leq t\right] + (1-t) \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array}. \tag{A.2}$$

A further generalization of the threshold lemma is the following, which is also useful in our analysis. (See [33, Lemma A.5].)

LEMMA A.4. *For any partition $X = X_1 \cup X_2$,*

$$F(\mathbf{x}) \geq \mathbf{E}\left[f((T_{>\lambda_1}(\mathbf{x}) \cap X_1) \cup (T_{>\lambda_2}(\mathbf{x}) \cap X_2))\right]$$

*where $\lambda_1, \lambda_2$ are independent and uniformly random in $[0,1]$.*