Nonmonotone Submodular Maximization via a Structural Continuous Greedy Algorithm

(Extended Abstract)

Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz

Computer Science Dept., Technion, Haifa, Israel
{moranfe,naor,schwartz}@cs.technion.ac.il

Abstract. Consider a situation in which one has a suboptimal solution S to a maximization problem which only constitutes a weak approximation to the problem. Suppose that even though the value of S is small compared to an optimal solution OPT to the problem, S happens to be structurally similar to OPT. A natural question to ask in this scenario is whether there is a way of improving the value of S based solely on this information. In this paper we introduce the *Structural Continuous Greedy Algorithm* which answers this question affirmatively in the context of the Nonmonotone Submodular Maximization Problem. Using this algorithm we are able to improve on the best approximation factor known for this problem.
In the Nonmonotone Submodular Maximization Problem we are given a non-negative submodular function f, and the objective is to find a subset maximizing f. This is considered one of the basic submodular optimization problems, generalizing many well known problems such as the Max Cut problem in undirected graphs. The current best approximation factor for this problem is 0.41 given by Gharan and Vondrák. On the other hand, Feige et al. showed that no algorithm can give a $0.5 + \varepsilon$ approximation for it (for any $\varepsilon > 0$). Our method yields an improved 0.42-approximation for the problem.

1 Introduction

Consider a situation in which one has a suboptimal solution S to a maximization problem. The most natural measure of the quality of S is its value with respect to the objective function. However, the structural similarity of S to an optimal solution OPT of the problem can be used as an additional measure. We are interested in the relation between these two measures. Suppose S is structurally similar to an optimal solution, yet its values is small. We call problems in which such sets exist *uncorrelated objective-structure problems*. For such problems it is possible for an algorithm to produce a solution which is structurally very similar to OPT, and yet is a poor approximation with respect to values. For example, in the Max Cut problem in undirected graphs, suppose the input is a star. If we put all vertices on one side of the cut, we get a solution which is only different from OPT in one vertex, yet its value is 0.

For every uncorrelated objective-structure problem $\mathcal{P}$, a natural question is whether there is an efficient algorithm that, given a solution S which is structurally similar to OPT, produces a solution S' of large value; i.e., we look for an algorithm that trades structural similarity to OPT for value. If we find such an algorithm for problem $\mathcal{P}$, then we can improve approximation algorithms for $\mathcal{P}$ which produce (directly, or indirectly) solutions that are structurally similar to OPT.

In this paper we suggest the *structural continuous greedy* algorithm that trades structural similarity to OPT for value in the context of the Nonmonotone Submodular Maximization Problem (NSM). Given a groundset $\mathcal{E}$, a function $f : 2^{\mathcal{E}} \to \mathbb{R}$ is called *submodular* if for every $A, B \subseteq \mathcal{E}$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. An instance of NSM consists of a non-negative submodular function f, and its objective is to find a subset of $\mathcal{E}$ maximizing f. We assume the *value oracle* model in which f is not given explicitly since its size might be exponential. Instead, the algorithm has access to an oracle that given a set S returns $f(S)$. This model is probably the most standard one, and is common throughout the literature.

To see why NSM is indeed an uncorrelated objective-structure problem, observe the following example. Consider the groundset $\mathcal{E} = [n] \times \{a, b\}$, and the submodular function: $f(S) = \sum_{i=1}^{n} g(S \cap \{(i, a), (i, b)\})$, where g is an indicator function for the event that its argument is a set of size 1. The set $\mathcal{E}' = [n/2] \times \{a, b\}$ has $f(\mathcal{E}') = 0$ although it contains half of the elements of every optimal solution.

In the next section we describe the structural continuous greedy algorithm. We need a few well known terms, which we define here for completeness. Given a function f, if $f(\varnothing) = 0$ we say that f is *normalized*, and if for every $A \subseteq B \subseteq \mathcal{E}$, $f(A) \leq f(B)$ (respectively, $f(A) \geq f(B)$), we say that f is *monotone* (respectively, *down monotone*). Also, we use the notation of OPT throughout the article to denote an optimal solution to the problem in question. If there are multiple optimal solutions, OPT denotes an arbitrary fixed one.

1.1 The Structural Continuous Greedy Algorithm

Let us formally define the structural similarity of a set S to OPT. One natural definition is $f(S \cap OPT)$. This definition makes sense when f is monotone. However, when f is allowed to be nonmonotone, we need a definition that also captures structural similarity due to elements outside of OPT that are also missing from S. Thus, we define the structural similarity of a set S to OPT as $f(S \cap OPT) + f(S \cup OPT)$[1].

Consider a set S. Removing all elements of $S - OPT$ changes the value of this set to $f(S \cap OPT)$; hence, if $f(S \cap OPT) > f(S)$, removing all the elements

[1] Note that the structural similarity of OPT to itself is $2f(OPT)$, and not $f(OPT)$ as one would expect. For simplicity, we avoid introducing a division by 2 to the definition. Also, the structural similarity depends on the specific optimal solution chosen. Hence, we can choose the optimal solution that fits us best, implying that the continuous greedy algorithm is guaranteed to improve a set if it is structurally similar to any optimal solution.

of $S - OPT$ from S increases the objective function. By submodularity, the last property also implies that there is at least one element in $S - OPT$ whose removal from S increases its value. Putting it all together, $f(S \cap OPT) > f(S)$ implies the existence of an element $e \in S - OPT$ such that $f(S - \{e\}) > f(S)$. Similarly, $f(S \cup OPT) > f(S)$ implies the existence of an element $e \in OPT - S$ such that $f(S \cup \{e\}) > f(S)$. We can now relate our observations to the definition of structural similarity: if $f(S \cap OPT) + f(S \cup OPT) > 2f(S)$, then the value of S can be improved by adding or removing a single element from it.

The last conclusion seems to suggest that given a set S which is structurally similar to OPT, yet has a low value, S can be improved using a local search algorithm that seeks at every stage to improve S by adding or removing a single element. However, as this algorithm adds and removes elements, the structural similarity to OPT might decrease; and we do not know how to relate this decrease to the increase in $f(S)$. Therefore, although we can conclude that such a local search algorithm will improve S, it is not clear how to give any positive lower bound on this improvement.

To work around this difficulty, we propose the structural continuous greedy algorithm. This algorithm maintains a fractional set (i.e., a set in which every element appears independently with some probability). Initially, every element of the input set S appears in the fractional set S' with probability 1, and every other element appears in it with probability 0. In every iteration the probability of each element can be changed by a small δ. The probability of an element is increased if doing so increases the expected value of the fractional set, assuming all other probabilities are unchanged; and decreased otherwise.

Using arguments similar to those above, we can lower bound the improvement in every iteration of the structural continuous greedy algorithm in terms of $\mathbb{E}[f(S' \cap OPT) + f(S' \cup OPT) - 2f(S')]$. Hence, as long as S' is structurally similar to OPT, in expectation, the structural continuous greedy is guaranteed to improve it by changing the probabilities of elements. Moreover, after m iterations, the change in the structural similarity of S' to OPT can be upper bounded using the following observation: every element of the original set S appears in S' with probability at least $1 - m\delta$, and every element outside of S appears in S' with probability at most $m\delta$; which intuitively means that S' is quite similar (structurally) to S, and therefore, also to OPT.

The structural continuous greedy inherits much of its structure from an algorithm of Vondrák [16] called "continuous greedy"; however, the two algorithms are analyzed very differently. The roots of [16]'s continuous greedy algorithm can be traced back to Wolsey [18], who also gives an algorithm called "continuous greedy", although, the two "continuous greedy" algorithms share only vague general ideas. The continuous greedy algorithm of [16] starts with an empty set, and then only increases the probabilities of elements. This is fine since [16] deals with monotone submodular functions. On the other hand, the structural continuous greedy, introduced in this paper, works with nonmonotone functions, which requires both modifications of the algorithm and its analysis (with respect to the algorithm and analysis of [16]).

There are two differences between our structural continuous greedy algorithm, and the "continuous greedy" algorithm of [16]. First, the initial point of the structural continuous greedy algorithm is an arbitrary set, and second, it may both *increase* and *decrease* probabilities. Though these differences are quite minor, the analysis of the two algorithms is completely different. The continuous greedy of [16] constructs a solution starting with an empty set, and its analysis strongly uses the monotonicity of f (which implies that increasing the probabilities of OPT's elements is always good). On the other hand, the structural continuous greedy improves existing sets, and its analysis is based on the structural similarity of the input to OPT.

1.2 Taking Advantage of the Structural Continuous Greedy

The structural continuous greedy algorithm can improve sets that are already structurally similar to OPT (at the cost of possibly decreasing their structural similarity to OPT). However, it is not clear, at first glance, how this can be used to produce an approximation algorithm for NSM. In order to answer this question, we have to understand the details of the known algorithms for NSM.

The first constant factor approximation algorithms for NSM were given by Feige et al. [4]. The simplest algorithm they provide simply selects every element of the ground set with probability 1/2. Feige et al. [4] showed that this simple algorithm already achieves a 1/4-approximation. They also suggested a local search algorithm which basically starts with an arbitrary set and adds or removes single elements as long as this improves the value of the set. The output set of this algorithm is called *locally optimal* because it cannot be improved by adding or removing a single element. In [4], it is shown that any locally optimal set gives a 1/3-approximation for NSM.

The problem with local search algorithms is that they tend to get "stuck" in local optima. One way to get around that is to add some noise to the system. For that purpose, [4] defines for every set S, a random set $R(S)$ that contains every element of S with probability 2/3 and every other element with probability 1/3 (in other words, to get $R(S)$, we start with S and switch the state of every element, from being in S to being outside of it or vice versa, with probability 1/3). The local search algorithm now tries to find a set that maximizes $\mathbb{E}[f(R(S))]$. Somewhat surprisingly, this randomized local search has an improved approximation ratio of 2/5. Looking more carefully at the analysis of [4], it actually shows that if S is a locally optimal set (with respect to the last algorithm), it has the following property:

$$\mathbb{E}[f(R(S))] + \frac{f(\bar{S})}{9} \geq \frac{4f(OPT)}{9} \quad .$$

This inequality guarantees that the expected values of $\max\{f(\bar{S}), f(R(S))\}$ is at least $0.4f(OPT)$. Now, observe that $f(S \cap OPT) + f(\bar{S} \cap OPT) \geq f(OPT)$, and $f(S \cup OPT) + f(\bar{S} \cup OPT) \geq f(OPT)$. This suggests that there is a negative correlation between the structural similarity to OPT of S and $\bar{S}$ in the following

sense: if one of them is structurally far from OPT, then the other one must be structurally close to OPT. Since $R(S)$ is basically S with some noise, we also get a negative correlation between the structural similarity to OPT of $R(S)$ and $\bar{S}$. Hence, if $R(S)$ has a low expected value, and it is too far from OPT for the structural continuous greedy algorithm to improve it significantly, then $\bar{S}$ must have both: significant value and structural similarity to OPT; hence, running the structural continuous greedy algorithm on it is guaranteed to produce a good set.

Using the above observations, we can get a 0.413-approximation, which already slightly improves on the state of the art algorithm of Gharan and Vondrack [5] that gives a 0.41-approximation (we defer details to a full version of this paper). However, we can do better if we use the structural continuous greedy algorithm together with [5]'s algorithm, though the combination of these two techniques requires some work. The algorithm of [5] is a *simulated annealing* algorithm. It starts as a local search algorithm with a lot of noise (i.e., $R(S)$ is obtained from S by adding or removing every element with high probability), and gradually decreases the noise level. Intuitively, starting with a lot of noise should help the algorithm avoid inferior local maxima.

Simulated annealing algorithms are common in practice, but they often turn out to be very hard to analyze. The analysis of [5] works as following. For some, relatively high, level of noise p_0, the algorithm is analyzed as a noisy local search algorithm, producing a lower bound on the value of the algorithm's solution at noise level p_0. Now, observe what happens as the algorithm reduces the noise level. In a locally optimal solution, infinitesimally increasing the probability of every element inside the solution, and decreasing the probability of every element outside of the solution, should only improve the solution. This change in the probabilities is exactly equivalent to a reduction in the noise level. Hence, whenever the algorithm reduces the noise level, since the current solution is locally optimal (the algorithm reduces the noise level only when it cannot improve the set by adding or removing a single element), the reduction in the noise level can only improve the solution.

The main observation of [5] is that it is possible to show a *positive* lower bound on the improvement achieved when the noise is reduced. This lower bound is negatively correlated with $f(\bar{S})$, where S is the current solution of the algorithm. This immediately implies that either the set $\bar{S}$ is good at some point, or the value of S significantly increases as the noise level decreases. Since we have a lower bound on the value of S at noise level p_0, an approximation guarantee follows.

The above description does not give an obvious way to combine [5]'s algorithm with the structural continuous greedy. In order for the structural continuous greedy algorithm to be useful, we must find sets in the proof of [5]'s algorithm with the following properties.

- The set is structurally similar to OPT, at least when [5]'s algorithm behaves poorly.
- The set has large value, again, at least when [5]'s algorithm behaves poorly.

If we find such a set, it can be improved using the structural continuous greedy algorithm, and then be used as an alternative solution. In other words, whenever the original algorithm behaves poorly, the structural continuous greedy produces for us a good alternative solution, which results in an improved approximation ratio.

To this end, we make the following observation. The bound, given by [5], on the improvement achieved when the noise is reduced is actually positively correlated with $f(S \cap OPT) + f(S \cup OPT)$. The negative correlation of the bound with $f(\bar{S})$ follows since, by submodularity, $f(\bar{S}) \geq f(OPT) - f(S \cap OPT) - f(S \cup OPT)$. Another inequality that follows from f's submodularity is $f(\bar{S} \cap OPT) + f(\bar{S} \cup OPT) \geq 2f(OPT) - f(S \cap OPT) - f(S \cup OPT)$. Hence, the lower bound is also negatively correlated with the structural similarity of $\bar{S}$ to OPT. In conclusion, if at some point the value of the algorithm's solution does not improve fast enough, then at that point $\bar{S}$ has two properties: it has a relatively high value and it is structurally similar to OPT. Therefore, applying the structural continuous greedy algorithm to $\bar{S}$ is guaranteed to produce a good set. This is the intuition behind our 0.42 approximation for NSM.

We note that [4] showed that no algorithm making only a polynomial number of oracle queries gives better than 1/2-approximation for NSM. This hardness result holds even when f is symmetric, i.e., $f(S) = f(\mathcal{E} - S)$, in which case it is tight [4].

1.3 Related Work

It is well known that submodular functions can be minimized in polynomial time [14]. However, maximizing a submodular function turns out to be a much more difficult task. Maximization of nonnegative submodular functions generalizes well known NP-hard problems such as Max-Cut [6], and [4] proved it impossible to give better than 0.5-approximation for it using a polynomial number of oracle queries. Related problems ask to maximize a nonnegative submodular function subject to various combinatorial constraints on the sets that we are allowed to choose [7, 11, 12, 17] or minimize a submodular function subject to such constraints [15, 8, 9].

Another line of work deals with maximizing normalized monotone submodular functions, again, subject to various combinatorial constraints. A continuous greedy algorithm was given by Calinescu et al. [1] for maximizing a normalized monotone submodular function subject to a matroid constraint. Later, Lee et al. [12] gave a local search algorithm achieving $1/p - \varepsilon$ approximation for maximizing a normalized monotone submodular function subject to intersection of p matroids. Kulik et al. [10] showed a $1 - 1/e - \varepsilon$ approximation algorithm for maximizing a normalized monotone submodular function subject to multiple knapsack constraints. Recently, Chekuri et al. [3], gave a nonmonotone counterpart of the continuous greedy algorithm of [16]. This result improves several nonmonotone submodular optimization problems. Some of the above results were generalized by Chekuri et al. [2], which gives a dependent rounding for various polytopes, including matroid and matroid-intersection polytops. The advantage

of this rounding technique is that it guarantees strong concentration bounds for submodular functions.

Organization. Section 2 gives the formal definitions of NSM and the notation we use. In addition, this section also states a few useful known technical lemmata. Section 3 describes the structural continuous greedy algorithm, and Section 4 shows how to combine this algorithm with the simulated annealing algorithm of [5].

2 Preliminaries

Formally, in NSM, we are given a nonnegative submodular function $f : \mathcal{E} \to \mathbb{R}^+$. The objective is to find a subset $S \subseteq \mathcal{E}$ maximizing $f(S)$. We denote the size of the ground set $\mathcal{E}$ by n. In all algorithms presented, we assume n is larger than any given constant (if this is not the case, one can solve NSM optimally using an exhaustive search).

There are two well known extensions of a submodular function $f : \mathcal{E} \to \mathbb{R}^+$ to the hypercube $[0,1]^{\mathcal{E}}$. The first one is $F(x) : [0,1]^{\mathcal{E}} \to \mathbb{R}^+$, the *multilinear extension* introduced by [1]. For a given vector $z \in [0,1]^{\mathcal{E}}$, let $R(z)$ be a set containing every element $e \in E$ with probability z_e, independently. The multilinear extension of f is defined as $F(z) = \mathbb{E}[R(z)]$. This extensions is called multilinear because it can also be written as following:

$$F(z) = \sum_{S \subseteq \mathcal{E}} \left(\prod_{e \in S} z_e \cdot \prod_{e \notin S} (1 - z_e) \cdot f(S) \right) \quad .$$

The other known extension of submodular functions is the Lovász extension introduced in [13]. Define $T_\lambda(z)$ to be the set of elements whose coordinate in z is at least λ. The Lovász extension of a submodular function $f : \mathcal{E} \to \mathbb{R}^+$ is defined as $\hat{f}(z) = \int_0^1 f(T_\lambda(z))d\lambda$. This definition can also be interpreted in probabilistic terms as the expected value of f over the set $T_\lambda(z)$, where λ is uniformly selected from the range $[0,1]$. In this paper, the Lovász extension is used only to lower bound the multilinear extension. This is done using the following theorem.

Theorem 1 (Lemma A.4 in [17]). *Let $F(z)$ and $\hat{f}(z)$ be the multilinear and Lovász extensions of a submodular function f, respectively. Then, for every $z \in [0,1]^{\mathcal{E}}$, $F(z) \geq \hat{f}(z)$.*

We abuse notation, and write $F(z \cup S)$ to denote $\mathbb{E}[f(R(z) \cup S)]$, $F(z - S)$ to denote $\mathbb{E}[f(R(z) - S)]$ and so on. We also use the shorthand $\partial_e F(z)$ for $F(z \cup e) - F(z - e)$. The multilinear nature of F yields the following useful observation which relates these notations.

Observation 1 *Let $F(z)$ be the multilinear extension of a submodular function $f : 2^{\mathcal{E}} \to R^+$. Then, for every $e \in \mathcal{E}$,*

$$\partial_e F(z) = \frac{F(z \cup e) - F(z)}{1 - z_e} = \frac{F(z) - F(z - e)}{z_e} \quad ,$$

assuming the denominators are not 0.

The following lemma shows that the change in $F(z)$ induced by making a small modification to the coordinates of z is almost equal to the sum of changes that would have resulted from modifying each coordinate independently. This kinds of lemmata are standard, however, we include its proof in Appendix A for completeness.

Lemma 1. *Consider two vectors $z, z' \in [0,1]^{\mathcal{E}}$ such that for every $e \in \mathcal{E}$, $|z_e - z'_e| \leq \delta$, for $\delta \leq n^{-3}$. Then, $F(z') - F(z) \geq \sum_{e \in \mathcal{E}} (z'_e - z_e) \cdot \partial_e F(z) - O(n^{-1}\delta)$.*

Given a set $S \subseteq E$ and a number $p \in [0.5, 1]$, let $z^p(S)$ be a vector of $[0,1]^{\mathcal{E}}$, containing p in the coordinate of every element of S, and $1-p$ in all other coordinates. Since the structural similarity of S to OPT is often used in our proofs, we denote it by $V(S)$, i.e., $V(S) = f(S \cap OPT) + f(S \cup OPT)$. For simplicity of the exposition we assume $f(OPT) = 1$. This assumption removes some clattering from the mathematical calculations, and can be easily removed.

3 The Structural Continuous Greedy Algorithm

In this section we show the structural continuous greedy algorithm for NSM. The structural continuous greedy algorithm improves a set S which have some structural resemblance to OPT (at the cost of possibly making the set less structurally similar to OPT).

Structural Continuous Greedy for NSM $(f, S, \mathcal{E})$**:**
1. Let $\delta = n^{-3}$. Initialize $t = 0$ and $z_0 = 1_S$.
2. While $t < 1$ do:
3. For every element $e \in \mathcal{E}$ do:
4. If $\partial_e F(z_t) > 0$,[a] $(z_{t+1})_e = \min\{1, (z_t)_e + \delta\}$.
5. If $\partial_e F(z_t) < 0$, $(z_{t+1})_e = \max\{0, (z_t)_e - \delta\}$.
6. $t \leftarrow t + \delta$
7. Return $R(z_1)$

[a] The algorithm does not have access to $\partial_e F(z_t)$, however, it can approximate it up to any required accuracy by averaging independent random samples. This is a standard practice, and we omit details (e.g., see [1]). Taking the error induced by the random sampling into account effects our results, with high probability, only by a lower order term.

We prove in this subsection the following theorem.

Theorem 2. *Assuming $V(S) \geq 2f(S)$, the Structural Continuous Greedy Algorithm for* NSM *produces a solution of expected value at least*

$$\frac{V(S)}{4} \cdot \left[2 - \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right] - O(n^{-1}) \ .$$

We define a few sets of elements that we will refer to. Let $\mathcal{E}_t^0$ (resp. $\mathcal{E}_t^1$) be the set of elements whose coordinates in z_t are 0 (resp. 1), $\mathcal{E}_t^+$ be the set $\{e \in \mathcal{E} | \partial_e F(z_t) \geq 0\}$, and $\mathcal{E}_t^-$ be $\mathcal{E} - \mathcal{E}_t^+$.

The following lemma shows that there is a set of elements for which the gain achieved by changing the coordinate of each one of them independently (i.e., when we change one coordinate, we assume the others are kept unchanged) is significant.

Lemma 2. *At every time t, $\sum_{e\in OPT\cap(\mathcal{E}_t^+-\mathcal{E}_t^1)}\partial_e(z_t)-\sum_{e\in\overline{OPT}\cap(\mathcal{E}_t^--\mathcal{E}_t^0)}\partial_e(z_t)\geq F(OPT\cap z_t)+F(OPT\cup z_t)-2F(z_t)$.*

Proof. Observe the following:

$$\sum_{e\in OPT\cap(\mathcal{E}_t^--\mathcal{E}_t^1)}F(z_t\cup e)-F(z_t)=\sum_{e\in OPT\cap(\mathcal{E}_t^--\mathcal{E}_t^1)}(1-(z_t)_e)\cdot\partial_eF(z_t)\leq 0\ .$$

Using this observation, we get:

$$\begin{aligned}\sum_{e\in OPT\cap(\mathcal{E}_t^+-\mathcal{E}_t^1)}\partial_eF(z_t)&\geq\sum_{e\in OPT\cap(\mathcal{E}_t^+-\mathcal{E}_t^1)}(1-(z_t)_e)\cdot\partial_eF(z_t)\\&=\sum_{e\in OPT\cap(\mathcal{E}_t^+-\mathcal{E}_t^1)}F(z_t\cup e)-F(z_t)\\&\geq\sum_{e\in OPT-\mathcal{E}_t^1}F(z_t\cup e)-F(z_t)\\&\geq F(z_t\cup(OPT-\mathcal{E}_t^1))-F(z_t)=F(z_t\cup OPT)\ .\end{aligned}\tag{1}$$

Analogously, we can also get:

$$\sum_{e\in\overline{OPT}\cap(\mathcal{E}_t^--\mathcal{E}_t^0)}\partial_eF(z_t)\leq F(z_t)-\cdot F(z_t-\overline{OPT})\ .\tag{2}$$

The lemma now follows by combining (1) and (2).

Together with Lemma 1, Lemma 2 shows that there is a set of elements whose coordinates can be changed together to produce a significant improvement. The following lemma shows that the algorithm indeed finds such a set.

Lemma 3. $F(z_{t+\delta})-F(z_t)\geq F(OPT\cap z_t)+F(OPT\cup z_t)-2F(z_t)-O(\delta n^{-1})$.

Proof. The algorithm increases the coordinates of all elements in $\mathcal{E}_t^+-\mathcal{E}_t^1$ by δ and decreases the coordinates of all elements in $\mathcal{E}_t^--\mathcal{E}_t^0$ by δ. Hence, by Lemma 1, we have:

$$F(z_{t+\delta})-F(z_t)\geq\delta\sum_{e\in\mathcal{E}_t^+-\mathcal{E}_t^1}\partial_eF(z)-\delta\sum_{e\in\mathcal{E}_t^--\mathcal{E}_t^0}\partial_eF(z)-O(n^{-1}\delta)\ .$$

The lemma now follows by combining this inequality with Lemma 2.

The following lemma lower bounds the similarity of $R(z_t)$ to OPT in terms of the similarity of the original set S to OPT and the time that passed so far.

Lemma 4. *For every time t, the following two inequalities hold:*

- $F(z_t \cap OPT) \geq (1-t) \cdot f(S \cap OPT)$.
- $F(z_t \cup OPT) \geq (1-t) \cdot f(S \cup OPT)$.

Proof. Clearly, for every element $e \in OPT$, $f(OPT) \geq f(OPT - e)$. Using submodularity, this implies that f is monotone on subsets of OPT. Observe that $f(X \cap OPT)$ is also a submodular function. Therefore, we can apply Theorem 1 to it, and get

$$F(z_t \cap OPT) \geq \int_0^1 f(T_\lambda(z_t) \cap OPT) d\lambda \geq \int_0^{1-t} f(T_\lambda(z_t) \cap OPT) d\lambda \ .$$

At time t, every element $e \in S$ must have $(z_t)_e \geq 1-t$, therefore, the set $T_\lambda(z_t)$ in the integrand must contain S. Hence, by the monotonicity of f over subsets of OPT, we get

$$F(z_t \cap OPT) \geq \int_0^{1-t} f(S \cap OPT) d\lambda = (1-t) \cdot f(S \cap OPT) \ .$$

The other inequality guaranteed by the lemma is proved analogously.

Plugging Lemma 4 into Lemma 3, we get the following lower bound on the improvement made by the algorithm at each step.

Corollary 1. $F(z_{t+\delta}) - F(z_t) \geq \delta[(1-t)[f(S \cap OPT) + f(S \cup OPT)] - 2F(z_t)] - O(n^{-1}\delta) = \delta[(1-t)V(S) - 2F(z_t)] - O(n^{-1}\delta)$.

Let g be the function which is the solution of the following recursive formula. $g(t+\delta) - g(t) = \delta[(1-t)V(S) - 2g(t)]$, with the boundary condition $g(0) = f(S)$. Lemmata 5 and 6 prove that it is sufficient to show that at some point $g(t) \geq X$ in order to prove $F(z_1) \geq X - O(n^{-1})$.

Lemma 5. *For evert time t, $F(z_t) \geq g(t) - O(n^{-1}t)$.*

Proof. Let c be the constant hiding behind the big O in Corollary 1. We prove by induction that $F(z_t) \geq g(t) - cn^{-1}t$. For time $t = 0$, the claim holds since $F(z_0) = f(S) = g(0)$. Assume that the claim holds for some time t, let us prove it for time $t + \delta$ using Corollary 1.

$$\begin{aligned} F(z_{t+\delta}) &\geq (1-t)\delta V(S) + (1-2\delta)F(z_t) - cn^{-1}\delta \\ &\geq (1-t)\delta V(S) + (1-2\delta)g(t) - cn^{-1}(\delta + t) = g(t+\delta) - cn^{-1}(\delta + t) \ . \end{aligned}$$

Lemma 6. *For every time t, $F(z_1) \geq F(z_t) - O(n^{-1})$.*

Proof. The algorithm increases only the coordinates of elements with positive $\partial_e F(z_t)$ and decreases only the coordinates of elements with negative $\partial_e F(z_t)$. Hence, by Lemma 1, for every time t, $F(z_{t+\delta}) \geq F(z_t) - O(n^{-1}\delta)$. Hence, the value of $F(z_t)$ can decrease only by $O(n^{-1}\delta)$ in every time step. Since there are only δ^{-1} time steps, the lemma follows.

Let h be the function $h(t) = \frac{3}{4}V(S) \cdot \left[1 - \frac{2t}{3} - e^{-2t}\right] + e^{-2t} \cdot f(S)$. Observe that $dh/dt = (1-t)V(S) - 2h(t)$. The next lemma shows that h lower bounds g (given some condition), and therefore, also lower bounds the value of the algorithm's solution.

Lemma 7. *If for every $t' \leq t$, $g(t') \leq 0.5(1-t)V(S)$, then $g(t) \geq h(t)$.*

Proof. We prove the lemma by induction on t. For time $t = 0$ the lemma follows from the definition of $h(0)$. Assume that the lemma holds for some time t, let us prove it for time $t + \delta$. Let t_1 be the last time in the range $[t, t+\delta]$ in which $h(t_1) \leq g(t)$.

$$\begin{aligned} h(t+\delta) &= h(t_1) + \int_{t_1}^{t+\delta} h'(\tau)d\tau = h(t_1) + \int_{t_1}^{t+\delta} \left((1-\tau)V(S) - 2h(\tau)\right) d\tau \\ &\leq g(t) + \int_{t_1}^{t+\delta} \left((1-t)V(S) - 2g(t)\right) d\tau \\ &= g(t) + (t+\delta-t_1)\left((1-t)V(S) - 2g(t)\right) \\ &\leq g(t) + \delta\left((1-t)V(S) - 2g(t)\right) = g(t+\delta) \ . \end{aligned}$$

Let $t^* = 0.5 \ln\left[3 - \frac{4f(S)}{V(S)}\right]$. Notice that if $f(S \cup OPT) + f(S \cap OPT) \geq 2f(S)$, then $\ell^* \in [0, 0.5 \ln 3] \subseteq [0,1]$.

Lemma 8. *Assuming $V(S) \geq 2f(S)$, $h(t^*) = 0.25V(S) \cdot \left[2 - \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right]$.*

Proof.

$$\begin{aligned} h(t^*) &= h\left(0.5 \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right) \\ &= \frac{3}{4}V(S) \cdot \left[1 - \frac{\ln\left[3 - \frac{4f(S)}{V(S)}\right]}{3} - \frac{V(S)}{3V(S) - 4f(S)}\right] + f(S) \cdot \frac{V(S)}{3V(S) - 4f(S)} \\ &= \frac{3}{4}V(S) \cdot \left[1 - \frac{\ln\left[3 - \frac{4f(S)}{V(S)}\right]}{3}\right] - \frac{V(S)}{4} = \frac{1}{4}V(S) \cdot \left[2 - \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right] \ . \end{aligned}$$

We are now ready to prove Theorem 2.

Proof (of Theorem 2). First let us claim that somewhere g gets the value:

$$0.25V(S) \cdot \left[2 - \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right] - O(n^{-3}) \ .$$

If the assumption of Lemma 7 does not hold for t^*, then g gets, somewhere, the value:

$$0.5(1-t^*)V(S) = 0.5\left(1 - 0.5 \ln\left[3 - \frac{4f(S)}{V(S)}\right]\right)V(S) \ .$$

Which proves the claim. Therefore, we can safely assume that the assumption of Lemma 7 holds, which implies that g gets value of at least $h(\delta \cdot \lfloor t^*/\delta \rfloor)$. In order to lower bound this value, we can use the observation that for every $t \in [0, t^*]$, $h'(t) \leq 2$. Hence,

$$h(\delta \cdot \lfloor t^*/\delta \rfloor) \geq h(t^*) - 2\delta \geq h(t^*) - 2n^{-3} \ .$$

And the theorem now follows from Lemmata 5 and 6.

4 Simulated Annealing Algorithm

Given a set S, let us denote by $C(S)$ the random set resulting from running the structural continuous greedy algorithm on S. Consider the following algorithm.

Simulated Annealing Algorithm for NSM $(f, \mathcal{E})$**:**
1. Let $d = 0.752 - \sqrt{2}/(1+\sqrt{2})$ and $\delta = d\lceil n^3 d \rceil^{-1}$.[a]
2. Initialize $p = \sqrt{2}/(1+\sqrt{2})$ and $A_p = \varnothing$.
3. Repeat:
4. Set $B_p \leftarrow A_p$
5. While there exists a set S such that:
   (i) $|S \oplus B_p| = 1$
   (ii) $F(z^p(S)) > F(z^p(B_t))$
6. Replace B_p with S
7. Set $A_{p+\delta} \leftarrow B_p$
8. Update $p \leftarrow p + \delta$
9. Until $p = 0.752$.
10. Return the best set in $\{R(z^{0.752}(B_{0.752}))\} \cup \{\bar{B}_p, C(\bar{B}_p)\}_{p=\frac{\sqrt{2}}{1+\sqrt{2}}}^{0.752}$

[a] Informally δ is the inverse of a number which is both at least n^3 and dividable by d.

Remark: As written the Simulated Annealing algorithm does not run in polynomial time for two reasons: the number of iterations that the algorithm makes might be exponential, and checking condition exactly *ii* cannot be done in polynomial time using only value oracle access to f. However, both problems can be solved using standard means (see, e.g., [12, 1]), at the cost of losing a lower order term in the approximation ratio. We omit the details.

Theorem 3. *The Simulated Annealing Algorithm for* NSM *returns a solution whose expected value is at least* 0.42.

The proof of the theorem combines ideas from the proof of the simulated annealing algorithm of [5] with the new observations described in Section 1.2. Due to lack of space, this proof is differed to Appendix B.

References

1. Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a monotone submodular function subject to a matroid constraint. To appear in SIAM Journal on Computing.

2. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*, 2010.
3. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *to appear in STOC '11*, 2011.
4. Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. In *FOCS*, pages 461–471, 2007.
5. Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. To appear in SODA 2011.
6. Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.
7. Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. To appear in WINE 2010.
8. Satoru Iwata and Kiyohito Nagano. Submodular function minimization under covering constraints. In *FOCS*, pages 671–680, 2009.
9. Stefanie Jegelka and Jeff Bilmes. Cooperative cuts: Graph cuts with submodular edge weights. Technical Report 189-03-2010, Max Planck Institute for Biological Cybernetics, Tuebingen, 2010.
10. Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *SODA*, pages 545–554, 2009.
11. Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing non-monotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, (4):2053–2078, 2010.
12. Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. In *APPROX*, pages 244–257, 2009.
13. László Lovász. Submodular functions and convexity. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: the State of the Art*, pages 235–257. Springer, 1983.
14. L. Lovász M. Grötschel and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatoria*, 1(2):169–197, 1981.
15. Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. In *FOCS*, pages 697–706, 2008.
16. Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.
17. Jan Vondrák. Symmetry and approximability of submodular maximization problems. In *FOCS*, pages 651–670, 2009.
18. L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

A Proof of Lemma 1

First, we need a few definitions. Let A be the set of elements e with $z'_e > z_e$, and let B be the set of elements with $z'_e < z_e$. For the sake of the proof, we assume $R(z')$ is formed from $R(z)$ using the following process. Every element of $A - R(z)$ is added to a set D with probability of $1 - (1 - z'_e)/(1 - z_e)$, and every element of $B \cap R(z)$ is added to D with probability $1 - z'_e/z_e$. Then, $R(z')$

is defined as $R(z) \oplus D$. Observe that every element $e \in \mathcal{E}$ gets into D with probability $|z_e - z'_e| \leq \delta$, independently. We are now going to bound the value of $F(z') - F(z) = \mathbb{E}[f(R(z')) - f(R(z))]$, given various constraints on D.

Lemma 9. $\sum_{e\in\mathcal{E}}(z'_e - z_e) \cdot \partial_e F(z) \leq \sum_{e\in\mathcal{E}} \Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] + 2n^{-1}\delta$.

Proof. Let $\mathcal{E}^+$ be the set of elements from $\mathcal{E}$ which have $(z'_e - z_e) \cdot \partial_e F(z) \geq 0$. Observe that for every $e \in \mathcal{E}^+$,

$$\begin{aligned}
(z'_e - z_e) \cdot \partial_e F(z) &= |z'_e - z_e| \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \\
&= \frac{\Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}]}{\prod_{e'\in\mathcal{E}-e}(1 - |z'_{e'} - z_{e'}|)} \\
&\leq \frac{\Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}]}{\prod_{e'\in\mathcal{E}-e}(1 - \delta)} \\
&= (1-\delta)^{1-|\mathcal{E}|} \cdot \Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \\
&< (1-\delta)^{-n} \cdot \Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \ .
\end{aligned}$$

And the term, $(1-\delta)^n$ can be lower bounded as following.

$$(1-\delta)^n \geq (1-n^{-3})^n \geq \sqrt[n^2]{e^{-1}(1-n^{-3})} \geq e^{-2n^{-2}} \geq 1 - 2n^{-2} \ .$$

Also, for every $e \in \mathcal{E} - \mathcal{E}^+$,

$$\begin{aligned}
(z'_e - z_e) \cdot \partial_e F(z) &= |z'_e - z_e| \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \\
&= \frac{\Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}]}{\prod_{e'\in\mathcal{E}-e}(1 - |z'_{e'} - z_{e'}|)} \\
&\leq \Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \ .
\end{aligned}$$

Combining everything, we get:

$$\begin{aligned}
\sum_{e\in\mathcal{E}}(z'_e - z_e) \cdot \partial_e F(z) - 2n^{-1}\delta &\leq \sum_{e\in\mathcal{E}}(z'_e - z_e) \cdot \partial_e F(z) - \delta n(2n^{-2}) \\
&\leq \sum_{e\in\mathcal{E}}[(z'_e - z_e) \cdot \partial_e F(z) - 2n^{-2} \cdot |(z'_e - z_e) \cdot \partial_e F(z)|] \\
&\leq \sum_{e\in\mathcal{E}} \Pr[D = \{e\}] \cdot \mathbb{E}[F(z') - F(z)|D = \{e\}] \ .
\end{aligned}$$

Lemma 10. $\mathbb{E}[F(z') - F(z_t) \mid D = \varnothing] = 0$.

Proof. $D = \varnothing$ implies $R(z') = R(z)$.

Lemma 11. $\Pr[|D| \geq 2] \cdot \mathbb{E}[F(z') - F(z) \mid |D| \geq 2] \geq -2n^{-1}\delta$.

Proof. Let us bound the probability that $|D| \geq 2$. Since every element enters into D with probability at most $\delta \leq n^{-3}$:

$$\begin{aligned}\Pr[|D| \geq 2] &\leq 1 - (1-\delta)^n - n \cdot \delta \cdot (1-\delta)^{n-1} \leq 1 - (1+n\delta) \cdot (1-\delta)^n \\ &\leq 1 - (1+n\delta) \cdot e^{-n\delta} \cdot (1-n\delta^2) \leq 1 - (1+n\delta)(1-n\delta)(1-n\delta^2) \\ &= 1 - (1-n^2\delta^2)(1-n\delta^2) \leq 2n^2\delta^2 \leq 2n^{-1}\delta \ .\end{aligned}$$

Therefore, we get:

$$\Pr[|D| \geq 2] \cdot \mathbb{E}[F(z') - F(z) \mid |D| \geq 2] \geq -\Pr[|D| \geq 2] \geq -2n^{-1}\delta \ .$$

Lemma 1 now follows immediately from the above lemmata and the law of total probability.

B Proof of Theorem 3

In this section we assume for the sake of contradiction that the Theorem 3 is wrong, i.e., the expected value of the simulated annealing's solution is less than 0.42.

Observation 2 *For every time p, $F(z^p(B_p)) \geq F(z^p(A_p))$.*

Proof. Clearly, the value of $F(z^p(B_p))$ never decreases. The observation now follows because the original value of $F(z^p(B_p))$ is $F(z^p(A_p))$.

Every set B_p is a local optimum, i.e., there is no set $S \subseteq \mathcal{E}$ such that:
(i) S is produced from B_p by adding or removing a single element.
(ii) $F(z^p(S)) > F(z^p(B_p))$ The following lemma applies to local optima.

Lemma 12 (Lemma 3.5 of [5]). *Let $q \in [1/3, 1/(1+\sqrt{2})]$, $p = 1-q$ and let S be a local optimum with respect to $F(z^p(S))$. Let $\beta = f(\bar{S})$. Then,*

$$F(z^p(S)) \geq \frac{1}{2}(1-q^2) - q(1-2q)\beta \ .$$

Corollary 2. *For $p' = \sqrt{2}/(1+\sqrt{2})$, $F(z^{p'}(B_{p'})) \geq 0.384$.*

Proof. By plugging $q = 1-p' = 1/(1+\sqrt{2})$ into Lemma 12, we get $F(z^{p'}(B_{p'})) \geq \sqrt{2}-1-(5\sqrt{2}-7)f(\bar{B}_{p'})$. Since we assumed Theorem 3 is wrong, we get $f(\bar{B}_{p'}) < 0.42$, and plugging this into the previous inequality yields

$$F(z^{p'}(B_{p'})) \geq \sqrt{2} - 1 - (5\sqrt{2}-7) \cdot 0.42 \geq 0.384 \ .$$

The next step is to show that every set B_p is structurally similar to OPT.

Lemma 13. *For $x \geq 2y \geq 0$, the expression $E = x \cdot \left[2 - \ln\left[3 - \frac{4y}{x}\right]\right]$ is an increasing function of x.*

Proof. The derivative of E by x is

$$\frac{dE}{dx} = \left[2 - \ln\left[3 - \frac{4y}{x}\right]\right] - x \cdot \frac{\left[3 - \frac{4y}{x}\right]'}{3 - \frac{4y}{x}}$$
$$= \left[2 - \ln\left[3 - \frac{4y}{x}\right]\right] - x \cdot \frac{\frac{4y}{x^2}}{3 - \frac{4y}{x}} = 2 - \ln\left[3 - \frac{4y}{x}\right] - \frac{\frac{4y}{x}}{3 - \frac{4y}{x}} \ .$$

Replacing $4y/x$ by z, and deriving the derivative by z, we get

$$\left[2 - \ln[3 - z] - \frac{z}{3 - z}\right]' = \frac{1}{3 - z} - \frac{(3 - z) + z}{(3 - z)^2} = \frac{3 - z - 3}{(3 - z)^2} \leq 0 \ .$$

Hence, increasing the value of z decreases the derivative. Therefore,

$$\frac{dE}{dx} \geq 2 - \ln\left[3 - \frac{4y}{2y}\right] - \frac{\frac{4y}{2y}}{3 - \frac{4y}{2y}} = 2 - \ln[3 - 2] - \frac{2}{3 - 2} = 2 - \ln 1 - 2 = 0 \ .$$

Lemma 14. *For every p, $V(B_p) \geq 0.689$.*

Proof. Since we assumed Theorem 3 is wrong, $f(\bar{B}_p), f(C(\bar{B}_p)) < 0.42$. Assume, also, $f(B_p \cup OPT) + f(B_p \cap OPT) = V(B_p) < 0.689$. Then:

- $f(\bar{B}_p) \geq 1 - [f(B_p \cup OPT) + f(B_p \cap OPT)] \geq 0.311$.
- $V(\bar{B}_p) = f(\bar{B}_p \cup OPT) + f(\bar{B}_p \cap OPT) \geq 2 - [f(B_p \cup OPT) + f(B_p \cap OPT)] \geq 1.311$.

We know $2f(\bar{B}_p) \leq 0.84 \leq 1.311$, hence, we can use Theorem 2 and Lemma 13.

$$C(\bar{B}_p) \geq 0.25V(\bar{B}_p) \cdot \left[2 - \ln\left[3 - \frac{4f(\bar{B}_p)}{V(\bar{B}_p)}\right]\right] - O(n^{-1})$$
$$\geq 0.25 \cdot 1.311 \cdot \left[2 - \ln\left[3 - \frac{4 \cdot 0.311}{1.311}\right]\right] - O(n^{-1}) \geq 0.42 \ .$$

Which contradicts the assumptions that Theorem 3 is wrong.

We now use the knowledge that B_p is structurally similar to OPT in order to lower bound the improvement the algorithm achieves when p increases.

Lemma 15. *For every p,*

$$(1 - p) \cdot [F(z^{p+\delta}(A_{p+\delta})) - F(z^p(B_p))] \geq$$
$$(1 - p)\delta \left(\sum_{e \in B_p \cap OPT} \partial_e F(z^p(B_p)) - \sum_{e \in \bar{B}_p - OPT} \partial_e F(z^p(B_p)) \right)$$
$$- p\delta \left(\sum_{e \in B_p - OPT} \partial_e F(z^p(B_p)) - \sum_{e \in \bar{B}_p \cap OPT} \partial_e F(z^p(B_p)) \right) - O(n^{-1}\delta) \ .$$

Proof. Since $A_{p+\delta} = B_p$, the coordinate of every element of B_p is larger in $z^{p+\delta}(A_{p+\delta})$ by δ in comparison to its value in $z^p(B_p)$. Similarly, the elements of $\bar{B}_p$ are smaller by δ. Hence, by Lemma 1,

$$F(z^{p+\delta}(A_{p+\delta})) - F(z^p(B_p)) \geq \delta \sum_{e \in B_p} \partial_e F(z^p(B_p)) - \delta \sum_{e \notin B_p} \partial_e F(z^p(B_p)) - O(n^{-1}\delta) \ . \tag{3}$$

Since F is multilinear, for every element $e \in B_p$ with $\partial_e F(z^p(B_p)) < 0$, removing e from B_p will increase the value of $F(z^p(B_p))$, which contradicts our assumption that B_p is locally optimal. Hence, for every $e \in B_p$, $\partial_e F(z^p(B_p)) \geq 0$. Similarly, we can also get that for every $e \notin B_p$, $\partial_e F(z^p(B_p)) \leq 0$. Therefore,

$$\sum_{e \in B_p - OPT} \partial_e F(z^p(B_p)) - \sum_{e \in \bar{B}_p \cap OPT} \partial_e F(z^p(B_p)) \geq 0 \ . \tag{4}$$

The lemma follows by multiplying (3) by $(1-p)$ and combining it with (4).

Lemma 16. *For every p,*

$$(1-p) \cdot [F(z^{p+\delta}(A_{p+\delta})) - F(z^p(B_p))] \geq \delta[2(1-p) + (2p-1)V(B_p) - 2F(z^p(B_p))] \ .$$

Proof. Observe the following inequality.

$$\begin{aligned}
&(1-p) \cdot \left[\sum_{e \in B_p \cap OPT} \partial_e F(z^p(B_p)) - \sum_{e \in \bar{B}_p - OPT} \partial_e F(z^p(B_p)) \right] \\
&- p \left[\sum_{e \in B_p - OPT} \partial_e F(z^p(B_p)) - \sum_{e \in \bar{B}_p \cap OPT} \partial_e F(z^p(B_p)) \right] \\
&= \sum_{e \in B_p \cap OPT} F(z^p(B_p) \cup e) - F(z^p(B_p)) - \sum_{e \in \bar{B}_p - OPT} F(z^p(B_p)) - F(z^p(B_p) - e) \\
&\quad - \sum_{e \in B_p - OPT} F(z^p(B_p)) - F(z^p(B_p) - e) + \sum_{e \in \bar{B}_p \cap OPT} F(z^p(B_p) \cup e) - F(z^p(B_p)) \\
&\geq F(z^p(B_p) \cup OPT) - F(z^p(B_p)) - F(z^p(B_p)) + F(z^p(B_p) - \overline{OPT})] \\
&\overset{(*)}{\geq} 2(1-p) + (2p-1)V(B_p) - 2F(z^p(B_p)) \ .
\end{aligned}$$

Where $(*)$ follows from Theorem 1. Plugging this inequality into Lemma 15 completes the proof.

Corollary 3. *For every p, $(1-p) \cdot \left[F(\bar{z}^{t+\delta}(B_{t+\delta})) - F(\bar{z}^t(B_t))\right] \geq \delta[1.317 - 0.634p - 2F(z^p(B_p))] - O(n^{-1}\delta)$.*

Proof. Follows immediately from Lemmata 16 and 14.

Let g be the function which is the solution of the following discrete recursive formula. $(1-p)\cdot[g(p+\delta)-g(p)] = \delta[(1.317-0.634p)f(OPT)-2g(p)]$, with the boundary condition $g(\sqrt{2}/(1+\sqrt{2})) = 0.384$.

Lemma 17. *For every p, $F(z^p(B_p)) \geq g(p) - O(pn^{-1}) = g(p) - O(n^{-1})$.*

Proof. Let c be the constant hiding behind the big O notation in Corollary 3. We prove by induction that $F(z^p(B_p)) \geq g(p) - 5cpn^{-1}$. For $p = \sqrt{2}/(1+\sqrt{2})$ the claims follows immediately from Corollary 2. Assume that the claim holds for some p, let us prove it for $p+\delta$.

$$\begin{aligned}
F(z^{p+\delta}(B_{p+\delta})) &\geq F(z^p(B_p)) + \frac{\delta[1.317-0.634p-2F(z^p(B_p))]-cn^{-1}\delta}{1-p} \\
&= F(z^p(B_p))\cdot\left[1-\frac{2\delta}{1-p}\right] + \frac{\delta(1.317-0.634p)-cn^{-1}\delta}{1-p} \\
&\geq \left[g(p)-5cpn^{-1}\right]\cdot\left[1-\frac{2\delta}{1-p}\right] + \frac{\delta(1.317-0.634p)-cn^{-1}\delta}{1-p} \\
&\geq g(p) + \frac{\delta[1.317-0.634p-2g(p)]}{1-p} - 5c(p+\delta)n^{-1} \\
&= g(p+\delta) - 5c(p+\delta)n^{-1}
\end{aligned}$$

Let h be the function:

$$h(p) = \frac{0.0405 - \frac{0.634\sqrt{2}}{1+\sqrt{2}}}{\frac{2}{5+2\sqrt{2}}}(1-p)^2 + 0.634(1-p) + 0.3435 \ .$$

Observe that h if a solution of the differential equation $(1-p)\cdot\frac{dh}{dp} = 1.317 - 0.634p - 2h(p)$. The maximum of h is attained at about $p = 0.752$, and $h(p)$ is monotonically increasing in the range $[\sqrt{2}/(1+\sqrt{2}), 0.752]$. Moreover, $h(0.752) > 0.42$, hence, it will be enough if we show that $g(p)$ is close to $h(p)$.

Lemma 18. *For $p \leq 0.752$, $g(p) \geq h(p)$.*

Proof. We prove the lemma by induction on p. For $p = \sqrt{2}/(1+\sqrt{2})$, $g(p) = h(p)$, and therefore, the lemma holds. Assume now that the lemma holds for some p, we will prove it for $p+\delta$. Observe the derivative of $h(p)$ is decreasing, hence,

$$\begin{aligned}
h(p+\delta) &= h(p) + \int_p^{p+\delta} h'(\rho)d\rho \leq h(p) + \int_p^{p+\delta} h'(p)d\rho = h(p) + \delta\cdot h'(p) \\
&= h(p) + \delta\cdot\frac{1.317-0.634p-2h(p)}{1-p} \\
&= h(p)\cdot\left(1-\frac{2\delta}{1-p}\right) + \delta\cdot\frac{1.317-0.634p}{1-p} \\
&\leq g(p)\cdot\left(1-\frac{2\delta}{1-p}\right) + \delta\cdot\frac{1.317-0.634p}{1-p} = g(p+\delta) \ .
\end{aligned}$$

We are now ready to complete the proof of prove Theorem 3 by proving that our assumption that it is wrong leads to a contradiction. The value of $F(z^p(B_p))$ at $p = 0.752$ is:

$$F(z^p(B_p)) \geq g(0.752) - O(n^{-1}) \geq h(0.752) - O(n^{-1}) > 0.42 \ .$$