Stability and Recovery for Independence Systems*

Vaggos Chatziafratis[1], Tim Roughgarden[†2], and Jan Vondrak[3]

1 **Stanford University, Computer Science Department, Stanford, CA, USA**
`vaggos@stanford.edu`
2 **Stanford University, Computer Science Department, Stanford, CA, USA**
`tim@stanford.edu`
3 **Stanford University, Department of Mathematics, Stanford, CA, USA**
`jvondrak@stanford.edu`

Abstract

Two genres of heuristics that are frequently reported to perform much better on "real-world" instances than in the worst case are *greedy algorithms* and *local search algorithms*. In this paper, we systematically study these two types of algorithms for the problem of maximizing a monotone submodular set function subject to downward-closed feasibility constraints. We consider *perturbation-stable* instances, in the sense of Bilu and Linial [11], and precisely identify the stability threshold beyond which these algorithms are guaranteed to recover the optimal solution. Byproducts of our work include the first definition of perturbation-stability for non-additive objective functions, and a resolution of the worst-case approximation guarantee of local search in p-extendible systems.

1998 ACM Subject Classification I.1.2 Analysis of Algorithms, G.2.1 Combinatorial Algorithms

Keywords and phrases Submodular, approximation, stability, Local Search, Greedy, p-systems

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.26

1 Introduction

Designing polynomial-time approximation algorithms with worst-case guarantees is one of the most common approaches to coping with NP-hard optimization problems. For many problems, even the best-achievable worst-case guarantee (assuming $P \neq NP$) is too weak to be immediately meaningful. Fortunately, it has been widely observed that most approximation algorithms typically compute solutions that are much better than their worst-case approximation guarantee would suggest (e.g. [17, 37]). Is there a mathematical explanation for this phenomenon?

One line of work addresses this question by restricting attention to instances that satisfy a *stability* condition, stating that there should be a "sufficiently prominent" optimal solution. Such conditions are analogs of the "large margin" assumptions that are often made in machine learning theory. Such assumptions reflect the belief that the instances arising in practice are ones that have a "meaningful solution". For example, if we run a clustering algorithm on a data set, it's because we're expecting that a "meaningful clustering" exists. The hope is that formalizing the assumption of a "meaningful solution" imposes additional structure on an instance that provably makes the problem easier than on worst-case instances.

Several such stability notions have been studied. In this work, we focus on the most well-studied one, that of *perturbation-stability* introduced by Bilu and Linial [11]. The idea

* Omitted proofs can be found in the full version of this paper: `https://arxiv.org/abs/1705.00127v1`.

† TR was supported by NSF award CCF-1524062.

25th Annual European Symposium on Algorithms (ESA 2017).
Editors: Kirk Pruhs and Christian Sohler; Article No. 26; pp. 26:1–26:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

behind the definition is that the optimal solution should be robust to small changes in the input (e.g., the edge weights of a graph). For if this is not true, then a minor misspecification of the data (which is often noisy in practice, anyways) can change the output of the algorithm. In data analysis, one is certainly hoping that the conclusions reached are not sensitive to small errors in the data. An informal definition of γ-perturbation-stability (henceforth simply γ-*stability*) is the following:

▶ **Definition 1** (γ-stability)**.** Given a weighted graph and an optimal solution S^* for some problem, we say that the instance is γ-*stable* if S^* remains the unique optimal solution, even when each edge weight is increased by an (edge-dependent) factor between 1 and γ.

Thus 1-stability is equivalent to the assumption that the optimal solution is unique. The bigger the γ, the stronger the assumption (since there are fewer instances we are required to solve), and hence, the easier the problem. The basic question is then **whether sufficiently stable instances of computationally hard problems are easier to solve**. The ultimate goal is to determine the *stability threshold* of a problem: the smallest value of γ such that the problem is polynomial-time solvable on γ-stable instances. We note that there is no general connection between hardness of approximation thresholds and stability thresholds of a problem – depending on the problem, each could be larger than the other (e.g., [7], where even though asymmetric k-center cannot be approximated to any constant factor, it can be solved optimally under 2-stability). Thus a good approximation algorithm need not recover an optimal solution in stable instances, and conversely.[1]

1.1 Our Results

Two genres of algorithms that are frequently reported to perform much better on "real-world" instances than in the worst case are *greedy algorithms* and *local search algorithms*. The goal of this paper is to systematically study these two types of algorithms through the lens of perturbation-stability. We carry this out for the rich and well-motivated class of problems that concern maximizing a monotone submodular set function subject to downward-closed feasibility constraints (as in e.g. [29, 34, 22]). Both greedy and local search algorithms can be naturally defined for all problems in this class. Special cases include [31] k-dimensional matching, asymmetric traveling salesman, influence maximization [24], welfare maximization in combinatorial auctions (with submodular valuations) [27, 41], and so on.

We organize our results along two different axes: whether the objective function is additive or submodular, and according to the "complexity" of the feasibility constraints. For the latter, we use the classic notions of the intersection of p matroids (for a parameter p), the more general notion of p-extendible systems (where a feasible solution can accommodate a new element after deleting at most p old ones), and the still more general notion of p-systems (where the cardinality of maximal independent sets can only differ by a p factor). Figure 1 summarizes our main results. We also prove that all of our results are tight.

Section 3 proves our results for the greedy algorithm in the case of additive objective functions. An interesting finding here is that for the most general set systems that we consider (p-systems), the greedy algorithm can have an infinite stability threshold, even

[1] For a silly example, consider an algorithm that checks if an instance is stable (by brute-force), if so returns the optimal solution (computed by brute force), and if not returns a terrible solution. Similarly, consider an α-approximation algorithm that uses brute force to always output a suboptimal solution, in every instance where one within α of optimal exists. For more natural (and polynomial-time) examples, see [7, 30].

TABLE 1: Additive Approximation

	Greedy	Local Search
p-Matroids	p	p
p-extendible	p	p^2 (new)
p-system	p	fails (new)

TABLE 2: Additive Recovery (new)

	Greedy	Local Search
p-Matroids	p	p
p-extendible	p	p^2
p-system	fails	fails

TABLE 3: Submodular Approximation

	Greedy	Local Search
p-Matroids	$p+1$	$p+1$
p-extendible	$p+1$	p^2+1 (new)
p-system	$p+1$	fails (new)

TABLE 4: Submodular Recovery (new)

	Greedy	Local Search
p-Matroids	$p+1$	$p+1$
p-extendible	$p+1$	p^2+1
p-system	fails	fails

Figure 1 Summary of old and new results. On the left we have previous approximation results about greedy and local search algorithms [25, 34, 22, 38] and our new local search approximation guarantees. (Each table entry indicates the worst-case approximation factor.) On the right are our recovery results for greedy and local search algorithms, with each table entry indicating the smallest γ such that the algorithm is optimal in every γ-stable instance. All of the results are tight.

though it is a good worst-case approximation algorithm. In fact, this crucial difference between approximation and stability also led us to give a different characterization of the p-extendible systems. Another interesting differentiation between stability and approximation shows up in the case of a uniform matroid (cardinality constraints).

Section 4 considers the greedy algorithm for maximizing a monotone submodular function. As all previous works on perturbation-stability have considered only problems with additive objective functions, here we need to formulate a notion of perturbation-stability for submodular functions, which boils down to defining the class of allowable perturbations of a submodular function f. The "sweet spot" – neither too restrictive nor too permissive – turns out to be the set of perturbed functions $\tilde{f}$ such that: (i) $\tilde{f}$ is also monotone and submodular; (ii) $\tilde{f}$ is a pointwise approximation of f ($\tilde{f}(S) \in [f(S), \gamma \cdot f(S)]$ for every S); and (iii) the marginal value of an element j with respect to a set S can only go up (in $\tilde{f}$), and by at most $(\gamma - 1)$ times the stand-alone value of j.[2] This definition specializes to the usual one in the special case of additive functions. Towards the end of this section, we also present an application for the welfare maximization problem, for which a weaker stability assumption is sufficient to guarantee recovery of the optimal allocation.

Section 5 identifies the smallest γ such that all local optima of γ-stable instances are also global optima, with both additive and submodular functions. A byproduct of our results here is new tight worst-case approximation guarantees for local search in p-extendible systems, which surprisingly were not known previously. The tight approximation guarantees are p^2 for additive functions and $p^2 + 1$ for monotone submodular functions.

[2] Each additional constraint on allowable perturbations $\tilde{f}$ weakens the stability assumption, resulting in a harder problem. For example, if one only assumes (i) and (ii) and not (iii), then the problem becomes "too easy", and every α-approximation algorithm automatically recovers the optimal solution in α-stable instances. If (iii) is replaced by the stronger condition that all marginal values change by a factor in $[1, \gamma]$, the problem becomes "too hard", with no positive recovery results possible (essentially because zero marginal values in f must stay zero in $\tilde{f}$).

1.2 Further Related Work

Perturbation-stability was defined by Bilu and Linial [11] in the context of the MaxCut problem. Subsequent work on perturbation-stability includes [10, 30, 2, 8, 7, 3, 39, 32, 6]. Independently of Bilu and Linial [11], Balcan, Blum and Gupta [5] introduced the related notion of *approximation stability* in the context of clustering problems like k-means and k-median. More technically distant analogs of these stability conditions (but with similar motivation) were proposed by [1, 19, 36]; see Ben-David [9] for further discussion.

2 Preliminaries

In this section we describe the notation and definitions which we use through the rest of the paper. We start by defining the family of p-systems and the problem of submodular maximization; then we present our two protagonist algorithms and the standard (additive) stability definition.

- p-Systems [25, 26]: Suppose we are given a (finite) ground set X of m elements (this could be the set of edges in a graph) and we are also given an *independence family* $\mathcal{I} \subseteq 2^X$, a family of subsets that is downward closed; that is, $A \in \mathcal{I}$ and $B \subseteq A$ imply that $B \in \mathcal{I}$. A set A is independent iff $A \in \mathcal{I}$. For a set $Y \subseteq X$, a set J is called a *base* of Y, if J is a maximal independent subset of Y; in other words $J \in \mathcal{I}$ and for each $e \in Y \setminus J$, $J + e \notin \mathcal{I}$. Note that Y may have multiple bases and that a base of Y may not be a base of a superset of Y. $(X, \mathcal{I})$ is said to be a p-system if for each $Y \subseteq X$ the following holds:

  $$\frac{\max_{J:J \text{ is a base of } Y} |J|}{\min_{J:J \text{ is a base of } Y} |J|} \leq p$$

  All set systems are assumed to be down-closed. There are some interesting special cases of p-systems [31, 12]:

  intersection of p matroids $\subseteq$ p-circuit-bounded systems $\subseteq$ p-extendible systems $\subseteq$ p-systems
- p-extendible: An independence system $(X, \mathcal{I})$ is p-extendible if the following holds: suppose we have $A \subseteq B$, $A, B \in \mathcal{I}$ and $A + e \in \mathcal{I}$; then there should exist a set $Z \subseteq B \setminus A$ such that $|Z| \leq p$ and $B \setminus Z + e \in \mathcal{I}$. We note here that p-extendible systems make sense only for integer values of p, whereas p-systems can have p being fractional and that 1-systems as well as 1-extendible systems are exactly matroids. It is a family of independence systems containing many important and seemingly unrelated problems like welfare maximization, k-dimensional Matching, Asymmetric Travelling Salesman Problem, weighted Δ-Independent Set (Δ: maximum degree) and others [31].
- Submodular Maximization: A set function $f : 2^X \to \mathbb{R}^+ \cup \{0\}$ is submodular if for every $A, B \subseteq X$, we have $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$. Given a p-system $(X, \mathcal{I})$ and a monotone submodular function f, we are interested in the problem of maximizing $f(S)$ over the independent sets $S \in \mathcal{I}$; in other words we wish to find $\max_{S \in \mathcal{I}} f(S)$. If f is additive, we can associate a weight w_e with each element $e \in X$ and we want to find $\max_{S \in \mathcal{I}} w(S)$, where $w(S) = \sum_{e \in S} w_e$.
- Greedy algorithm: It starts with $S = \emptyset$ and greedily picks elements of X that will increase its objective value by the most, while remaining feasible i.e. picks $e^* = \arg\max_{e \in X, S+e \in \mathcal{I}} (f(S + e) - f(S))$. It is a well-known fact [25, 34], that for any p-system, the standard greedy algorithm is a $(p+1)$-approximation (if f is additive, Greedy is a p-approximation).

- (p, q)-Local Search: It starts from a feasible solution and at each iteration seeks for an improving move. In particular, starting from any $S \in \mathcal{I}$, it tries to find a *better* $S' \in \mathcal{I}$ with: $|S \setminus S'| \leq p$, $|S' \setminus S| \leq q$ and $f(S') > f(S)$. If it finds such a feasible solution S', it switches to S' and repeats. It stops when no improving move can be made. Note that the stopping condition and its performance depend on the size of the (p, q)-*neighbourhood* used. We note that $(p, 1)$-local search is necessary for p-extendible systems. For recent improvements on Local Search performance in the case of matroids, we refer the reader to [26].
- Stable instances: Stability can be defined in general for instances of weighted optimization problems [11], where the objective function w is additive. In our case, given a p-system and an *additive* function w we wish to maximize over the p-system, we call the instance γ-stable, if the optimal solution $S^* \in \mathcal{I}$ remains the unique optimum, even after assigning a new weight $\tilde{w}_e$ to an element e such that $w_e \leq \tilde{w}_e \leq \gamma \cdot w_e$. In an extreme case, we can keep the weights of the elements in optimum the same and increase all others by a factor of γ; the optimum should remain the same. Sometimes, we say that we γ-perturb the input when we multiply some weights by at most γ. We will see in Section 4 how to extend this additive stability definition to stability for submodular functions.

3 Warm-up: Additive Case and Greedy Recovery

In this section, as a warm-up, we deal with additive functions, proving the first positive recovery result for the greedy algorithm and showing that it is tight.

3.1 Exact Recovery for p-extendible, p-stable systems

We are given an independence set system $(X, \mathcal{I}, w)$ and we want to find an independent solution $S^* \in \mathcal{I}$ with maximum weight, where for $I \in \mathcal{I} : w(I) = \sum_{e \in I} w(e)$. We are interested in the performance of the standard greedy algorithm and we can prove the following:

▶ **Theorem 2.** *Given an instance of a p-extendible independence system $(X, \mathcal{I}, w)$, that has a p-stable optimal solution $S^* = \arg\max_{I \in \mathcal{I}} w(I)$, the Greedy algorithm exactly recovers S^*.*

Proof. From the definition of p-extendibility we know that for the system $\mathcal{I}$, the following holds: suppose $A \subseteq B, A, B \in \mathcal{I}$ and $A + e \in \mathcal{I}$, then there is a set $Z \subseteq B \setminus A$ such that $|Z| \leq p$ and $B \setminus Z + e \in \mathcal{I}$. The Greedy starts from the empty set and greedily picks elements with maximum weight subject to being feasible; it finally outputs S which is a maximal solution, i.e. $S \cup \{e\} \notin \mathcal{I}, \forall e \in X \setminus S$. In order to get exact recovery, we want to show that $S \equiv S^*$.

Let's suppose $S \setminus S^* \neq \emptyset$. Then, out of all the elements of $S \setminus S^*$ that the Greedy selected, let's focus on the first element $e_1 \in S \setminus S^*$. Let $S_{\{e_1\}}$ denote the greedy solution right before it picked element e_1. Note that before choosing e_1, greedy $S_{\{e_1\}}$ was in agreement with the optimal solution, i.e. $S_{\{e_1\}} \subseteq S^*$. Since $e_1 \notin S^*$, we can use the p-extendibility, where we specify $A = S_{\{e_1\}}, B = S^*, e = e_1$ ($A + e \equiv S_{\{e_1\}} + e_1 \in \mathcal{I}$, since Greedy is always feasible) and we get, following the above definition, that there exists set of elements $Z \subseteq S^* \setminus A \equiv S^* \setminus S_{\{e_1\}}$, with $|Z| \leq p$ and $(S^* \setminus Z) \cup \{e_1\} \in \mathcal{I}$. This intuitively means that the element e_1 has conflicts with the elements in $Z \subseteq S^* \setminus S_{\{e_1\}}$, but if we remove at most $|Z| \leq p$ elements from $S^* \setminus S_{\{e_1\}}$, we get no conflicts and thus an independent (feasible) solution according to the system $\mathcal{I}$.

We call this solution J, i.e. $J = (S^* \setminus Z) \cup \{e_1\} \in \mathcal{I}$ (note $J \neq S^*$) and we will show that we can perturb the instance (new weight function $\tilde{w}$) no more than a factor of p, so that J's weight is at least that of the optimal, i.e. $\tilde{w}(J) \geq \tilde{w}(S^*)$, which would be a contradiction to the p-stability of the given instance. All we have to do is perturb the instance by multiplying the weight of the element e_1 by p. By the greedy criterion for picking elements (note that all elements of Z were available to Greedy at the point it chose e_1) and the fact that $|Z| \leq p$ we get:

$$\forall e \in Z \subseteq \left(S^* \setminus S_{\{e_1\}}\right) : w(e_1) \geq w(e) \implies p \cdot w(e_1) \geq \sum_{e \in Z} w(e) = w(Z) \tag{1}$$

which implies that the weight of the set J is actually no less than the weight of S^* in the aforementioned perturbed instance (weight function $\tilde{w}$). Indeed:

$$\tilde{w}(J) = \tilde{w}((S^* \setminus Z) \cup e_1) = \tilde{w}(S^* \setminus Z) + \tilde{w}(e_1) = w(S^*) - w(Z) + p \cdot w(e_1) \geq w(S^*) = \tilde{w}(S^*)$$

where for the last inequality we used (1). This is a contradiction because it violates the p-stability property (the optimal solution should stand out as the unique optimum for any p-perturbation) and thus we conclude that $S \setminus S^* = \emptyset$. Since Greedy outputs a maximal solution, we conclude that S coincides with S^* and so Greedy exactly recovers the optimal solution. ◀

We next show that our result is tight both in terms of the stability factor and the generality of p-extendible systems.

▶ **Proposition 3.** *There exist p-extendible systems with a $(p - \epsilon)$-stable optimal solution S^*, for which the Greedy fails to recover it.*

Proof. Take a Maximum Weight Matching instance (here $p = 2$): a path of length 3 with weights $(1, 1 + \epsilon', 1)$. The Greedy fails to recover the optimal solution S^*, since it picks the $(1 + \epsilon')$ edge whereas it should have picked both the other edges. For the right choice of ϵ' ($\epsilon' < \frac{\epsilon}{2-\epsilon}$), we can make the instance arbitrarily close to $(p - \epsilon) = (2 - \epsilon)$ stable. Observe that we can give such examples for any value of p (consider the p-dimensional Matching problem) and that the example can be made arbitrarily large just by repeating it. ◀

▶ **Proposition 4.** *There are p-systems whose optimal solution S^* is M-stable (for arbitrary $M > 1$) and for which the greedy algorithm fails to recover it.*

Proof. The example is based on a knapsack constraint. Fix $M' > 1$ and let the size of the knapsack $B = M' + 1$. We will have elements of type A ($|A| = M'$), a special element e^* and elements of type C ($|C| = M'$). The pair (value, size) for elements in A, C is respectively: $(2, 1), (1, \frac{1}{M'})$ and for $e^* : (1 + \epsilon, 1), \epsilon > 0$. Note that the optimal solution S^* is $A \cup C$ with total value $2M' + M' = 3M'$ and size $M' + M' \cdot \frac{1}{M'} = M' + 1$ (fits in the knapsack). However, Greedy will pick $A \cup \{e^*\}$ for a total value of $2M' + 1 + \epsilon$ and size $M' + 1$. Note that this is a p-system for a value of $p < 2$ since any feasible solution S can be extended to a solution S' with $|S'| \geq M' + 1$ and the largest feasible solution has $2M'$ elements (there are only $2M' + 1$ elements in total). However, this is not a 2-extendible system (it is actually an M'-extendible system) and we see that even if it is $(M' - 1)$-stable, Greedy still fails to recover the optimal solution S^*. To see why it is $(M' - 1)$-stable, note that the only γ-perturbation (perturbations are allowed only on the values, not the sizes) we can make to favour the greedy solution is to the element e^*, thus we would need $\gamma(1 + \epsilon) \geq M' \implies \gamma > M' - 1$ (ϵ is small). Choose $M' = M + 1$ and this concludes the proof. We also note that a variation of this counterexample would trick as well the (more natural) Greedy that sorts the elements according to value density ($\frac{v_i}{s_i}$) instead of just their value. ◀

We find Proposition 4 surprising, given that the greedy algorithm is a good worst-case approximation algorithm for such problems. The above "bad" example leads us to the definition of *hereditary* systems; it turns out that this is another characterization of the p-extendible systems. Due to space constraints, we give the formal definition in the full version (in Appendix B).

4 The Case of Submodular Functions

This section considers recovery results for stable instances where the objective function is monotone and submodular. Submodular functions are widely used in many areas ranging from mathematics to economics, and they model situations with *diminishing returns*. Famous examples include influence maximization [24, 33, 20, 13] and welfare maximization in auctions and game theory [27, 35]. For example, in influence maximization, the goal is to "activate" a subset of the participants in a social network (e.g., provide with information, or a promotional product) so as to maximize the expected spread of the idea or product. The diffusion of information is usually modeled with submodular functions (indicating the probability that a node adopts a new idea or product as a function of how many of her neighbors in the social network have already done so). In practice, the submodular functions in the input are estimated from data and hence are noisy (e.g. [4]). One hopes that the output of an influence maximization algorithm (which is typically a greedy algorithm [24]) is robust to modest errors in the specification of the submodular function. This section proposes a definition to make this idea precise, and proves tight results for greedy and local search algorithms under this stability notion.

4.1 Stability for submodular functions

All previous work on perturbation-stability considered only additive objective functions. We next state our extension to submodular functions.

▶ **Definition 5** (γ-perturbation, $\gamma \geq 1$)**.** Given a monotone submodular function $f : 2^X \to \mathbb{R}^+ \cup \{0\}$, we define $f_S(j) = f(S + j) - f(S)$. A γ-perturbation of f is any function $\tilde{f}$ such that the following three properties hold:

1. $\tilde{f}$ is monotone and submodular.
2. $f \leq \tilde{f} \leq \gamma f$, or in other words $f(S) \leq \tilde{f}(S) \leq \gamma f(S)$ for all $S \subseteq X$.
3. For all $S \subseteq X$ and $j \in X \setminus S$, $0 \leq \tilde{f}_S(j) - f_S(j) \leq (\gamma - 1) \cdot f(\{j\})$.

The definition of a γ-stable instance is then defined as usual.

▶ **Definition 6** (γ-stability)**.** Given an independence system $(X, \mathcal{I})$ and a monotone submodular function $f : 2^X \to \mathbb{R}^+ \cup \{0\}$, let $S^* := \arg\max_{S \in \mathcal{I}} f(S)$. The instance is γ-*stable* if for every γ-perturbation of the initial function f, S^* remains the unique optimal solution.

As discussed in the Introduction, while Definition 5 is perhaps not the first one that comes to mind, it appears to be the "sweet spot". Natural modifications[3] of the definition are generally either too restrictive (rendering the problem impossible, e.g. if property 3 is replaced with

[3] If we dropped Property 3, then *any* c-approximation algorithm ($c \geq 1$) returning a solution S with $f(S) \geq \frac{1}{c} \cdot f(S^*)$, could be made to have value equal with S^* in the c-perturbed version $\tilde{f}(S) = cf(S) \geq f(S^*) = \tilde{f}(S^*)$. If we dropped Property 2, then the definition would not be a generalization for the case of additive perturbations as we could have $\tilde{f}(S) > \gamma f(S)$ for some set S, because of the quantity $f(\{j\})$ in Property 3, which is relative to the *empty set* and may be large compared to $f_S(j)$.

relative perturbations since then the zero marginal values in f must stay zero in $\tilde{f}$) or too permissive (rendering the problem uninteresting, with all α-approximation algorithms equally good).

▶ **Proposition 7.** *Definition 5 specializes to perturbation-stability in the special case of an additive objective function.*

Proof. This follows easily since if the function f is additive, then there will be no dependence of the element's j marginal value on the current set S and thus property 3 from the above γ**-perturbation** definition just becomes:

$$0 \leq \tilde{f}_S(j) - f_S(j) \leq (\gamma-1)\cdot f(j) \iff 0 \leq \tilde{f}(j) - f(j) \leq (\gamma-1)\cdot f(j) \iff f(j) \leq \tilde{f}(j) \leq \gamma\cdot f(j)$$

which is exactly the standard notion of stability introduced by [11]. Note that this also implies the first condition for all sets S: $f(S) \leq \tilde{f}(S) \leq \gamma f(S)$, by the additivity of f. ◀

We now prove a useful proposition that we will often use when proving recovery results for submodular maximization. Informally, we show that multiplying the marginal improvements of the choices made by an algorithm by γ is a valid γ-perturbation.

▶ **Proposition 8.** *Let f be a monotone submodular function. Fix an ordered sequence of elements $e_1, e_2, \dots, e_k$, and let $\delta_i = f(\{e_1, \dots, e_i\}) - f(\{e_1, \dots, e_{i-1}\})$. Then $\tilde{f}$ defined by $\tilde{f}(S) := f(S) + (\gamma - 1)\sum_{i:e_i \in S} \delta_i$ is a valid γ-perturbation of f.*

Proof. Let us verify the conditions of a γ-perturbation.

First, $\tilde{f}$ is monotone submodular, since it is a sum of a monotone submodular and a monotone additive function ($\delta_i \geq 0$ by monotonicity).

Second, we have $f(S) \leq \tilde{f}(S) = f(S) + (\gamma-1)\sum_{i:e_i\in S} \delta_i \leq f(S) + (\gamma-1) f(S \cap \{e_1, \dots, e_k\})$ by submodularity, and by monotonicity this is at most $\gamma f(S)$.

Third, the marginal values of $\tilde{f}$ are $\tilde{f}_S(e_i) = f_S(e_i) + (\gamma - 1)\delta_i \leq f_S(e_i) + (\gamma - 1) f(\{e_i\})$ (and unchanged for elements other than the e_i). ◀

4.2 Greedy recovery and submodularity

The main result here is that the standard greedy algorithm can recover the optimal solution of a p-extendible system, if the optimal solution is $(p + 1)$-stable (as it was defined in Section 4.1).

▶ **Theorem 9** (Greedy Recovery)**.** *Given a monotone submodular function f to maximize over a p-extendible system $(X, \mathcal{I})$, if the optimal solution $S^* = \arg\max_{S\in\mathcal{I}} f(S)$ is $(p + 1)$-stable, then the greedy algorithm recovers S^* exactly.*

Proof. The proof generalizes the argument we used in the additive case so that we handle submodularity and the proving strategy resembles the proof of the approximation guarantee for the greedy algorithm for submodular maximization on p-extendible systems [12]. Let's denote by $S = \{e_1, \dots, e_k\}$ the solution produced by Greedy (in the order that Greedy picked them) and S^* the optimal solution. To give some intuition, in the additive case before, we used the property of p-extendibility in order to say that every element that appears in S but not in S^* could be "boosted" by a factor of p to obtain an even better optimal solution, which would be a contradiction, because of the p-stability. Now, due to submodularity, we need to be careful that we make this exchange argument in a cautious manner.

For $0 \le i \le k$, let $S_i = \{e_1, e_2, \dots, e_i\}$ denote the first i elements picked by Greedy (with $S_0 = \emptyset$). Let $\delta_i = f_{S_{i-1}}(e_i) = f(S_i) - f(S_{i-1})$. Using the p-extendibility property, we can find a chain of sets $S^* = T_0 \supseteq T_1 \supseteq \dots \supseteq T_k = \emptyset$ such that for $1 \le i \le k$:

$$S_i \cup T_i \in \mathcal{I}, \; S_i \cap T_i = \emptyset \text{ and } |T_{i-1} \setminus T_i| \le p.$$

The above means that every element in T_i is a candidate for Greedy in step $i+1$. We construct the chain as follows: Let $T_0 = S^*$; we show how to construct T_i from T_{i-1}:

1. If $e_i \in T_{i-1}$, we define $S_i^* = \{e_i\}$ and $T_i = T_{i-1} - e_i$. This corresponds to the trivial case when Greedy, at stage i, happens to choose an element e_i that also belongs to the optimal solution S^*.
2. Otherwise ($e_i \notin T_{i-1}$), we let S_i^* be a smallest subset of T_{i-1} such that $(S_{i-1} \cup T_{i-1}) \setminus S_i^* + e_i$ is independent and since $\mathcal{I}$ is p-extendible, we have $|S_i^*| \le p$. We let $T_i = T_{i-1} \setminus S_i^*$.

By the above definitions for S_i, T_i, S_i^* it follows that $S_i \cup T_i \in \mathcal{I}$ and $S_i \cap T_i = \emptyset$. By the maximality of Greedy (stopping condition: $\{e | S_k + e \in \mathcal{I}\} = \emptyset$) and the fact that $S_k \cup T_k \in \mathcal{I}$, it also follows that $T_k = \emptyset$. Since Greedy could have picked, instead of e_i, any of the elements in S_i^* (in fact T_{i-1}) we get: $\delta_i \ge \frac{1}{p} f_{S_{i-1}}(S_i^*)$ (recall that $|S_i^*| \le p$).

Let us assume now that the Greedy solution S is not optimal. We use Proposition 8 to define a $(p+1)$-perturbation that produces a new optimal solution. Let's suppose $|S \setminus S^*| = l$ and let's rename the elements e_i such that $|S \setminus S^*| = \{e_1, e_2, \dots, e_l\}$ in the order that the Greedy picked the elements. Then we define $\tilde{f}(T)$ for every T by $f(T) = f(T) + p \sum_{1 \le i \le l : e_i \in T} \delta_i$, where $\delta_i = f_{S_i}(e_i) = f(\{e_1, \dots, e_i\}) - f(\{e_1, \dots, e_{i-1}\})$. Using Proposition 8, this is a valid $(p+1)$-perturbation. For the greedy solution S, we obtain:

$$\tilde{f}(S) = f(S) + p \sum_{i=0}^{l-1} f_{S_i}(e_{i+1}) \ge$$

$$\ge f(S) + \sum_{i=0}^{l-1} f_{S_i}(S_{i+1}^*) \ge f(S) + \sum_{i=0}^{l-1} f_S(S_{i+1}^*) \ge f(S) + f_S(S^* \setminus S) =$$

$$= f(S) + (f((S^* \setminus S) \cup S) - f(S)) = f(S^* \cup S) \ge f(S^*) = \tilde{f}(S^*).$$

We ended up with $\tilde{f}(S) \ge \tilde{f}(S^*)$ which means that S^* is no longer the unique optimum and hence we get a contradiction to the $(p+1)$-stability of S^*. ◀

▶ Remark. If instead of exact access to the values of the function f, we had an α-approximate oracle, then the proof easily extends to handle this case as well. In particular, suppose each element e_i picked by Greedy at stage i satisfies $f_{S_{i-1}}(e_i) \ge \alpha \max_{e \in A_i} f_{S_{i-1}}(e)$, where A_i is the set of all candidate augmentations of S_{i-1}. Here $\alpha \le 1$. We would then have that the greedy marginal improvement $\delta_i \ge \frac{\alpha}{p} f_{S_{i-1}}(S_i^*)$ and thus we would need $\gamma - 1 = \frac{p}{\alpha}$ leading to exact recovery of $\left(\frac{p+\alpha}{\alpha}\right)$-stable instances ($\alpha \le 1$).

4.3 Welfare Maximization

In many situations, like the welfare maximization problem [27, 35, 41], the submodular function f we wish to maximize has a special form, e.g. it may be written as a sum of other submodular functions f_i (each of which may correspond to the player's i valuation on different allocations of the items). In this special case, we have $f(S) = \sum_{i=1}^{n} f_i(S)$ and from Theorem 9 Greedy recovers the optimal solution S^* for the case of matroids, which are 1-extendible, if S^* is 2-stable.

However, for *sum* functions $f = \sum_i f_i$, we may as well hope that a stronger recovery result is true, i.e. that Greedy recovers the optimal solution of $\max\{f(S) = \sum_i f_i : S \in \mathcal{I}\}$, where the optimum is 2-stable only with respect to 2-perturbations of the individual functions f_i. This is indeed true (for the proof, we refer the reader to Appendix A of the full version).

▶ **Theorem 10.** *Let $(X, \mathcal{I})$ be a matroid on the elements of X, let $B_1, B_2, \dots, B_k$ be a partition of X, $f_i : 2^{B_i} \to \mathbb{R}^+ \cup \{0\}$, for $i \in \{1, 2, \dots, k\}$ be monotone submodular and let $f = \sum_{i=1}^k f_i$. Suppose the optimal solution S^* of $\max\{f(S) : S \in \mathcal{I}\}$ is 2-stable only with respect to individual perturbations of the functions f_i. Then, Greedy recovers S^*.*

5 Local Search Performance

In this section we discuss *local search* [28] (described in Section 2). Local search often gives better results than Greedy, at the cost of a slower running time – for example for submodular maximization subject to the intersection of k matroids [26, 21], and for k-set packing [40, 18, 23]. For some interesting recent results about local search in *beyond-worst-case* settings and on geometric optimization we refer the reader to [14, 15, 16].

Somewhat surprisingly, it was not known (to our knowledge) how local search performs for p-systems and p-extendible systems. (We recall that the greedy algorithm gives a factor of $1/p$ for maximization of an additive function and $1/(p+1)$ for maximization of a monotone submodular function under these constraints.) Here, we prove that local search in fact performs worse than Greedy for these constraints. Although it gives a $1/p$-approximation for cardinality maximization under a p-system constraint (essentially by definition), it does not give any bounded approximation factor for additive function maximization under a p-system, and only a $1/p^2$-approximation under a p-extendible system.

5.1 Local search fails for p-systems

We construct simple examples where local search will not recover any fraction of the maximum-weight solution for p-systems (even if it is arbitrarily stable, $p = 2$, and even if we allow large exchange neighborhoods). In particular, consider a ground set $X = A \cup \{e^*\}$ where $|A| = n$. The independent sets of $\mathcal{I}$ are:

- any subset of A, or
- e^* plus any subset of at most $n/2$ elements of A.

Note that this is a 2-system, because for $S \subseteq X$, any independent subset of S can be extended to an independent set of size at least $\min\{|S|, n/2\}$, and the maximum independent subset of S has size at most $\min\{|S|, n\}$. The weights could be 0 on A, and 1 on the special element e^*. So the optimum is $w(e^*) = 1$ (observe that the optimal solution is c-stable for arbitrarily large c). However, A is a local optimum, unless we are willing to swap out $n/2$ elements, which is not possible for efficient local search.

5.2 Lower bound for p-extendible systems

Let us consider the following instance. Let $X = A \cup B$ where A, B are disjoint sets. We define $\mathcal{I} \subseteq 2^X$ as follows: $S \in \mathcal{I}$ iff

- $|S \cap A| + p|S \cap B| \leq |A|$, or
- $p|S \cap A| + |S \cap B| \leq |B|$.

▶ **Lemma 11.** *For any A, B disjoint, the above is a p-extendible system.*

Proof. Let $S \subseteq T$ and $i \in X \setminus T$ be such that $S + i \in \mathcal{I}$ and $T \in \mathcal{I}$. We need to prove that there is $Z \subseteq T \setminus S, |Z| \leq p$ such that $(T \setminus Z) + i \in \mathcal{I}$.

We can assume that $|T \setminus S| > p$, because otherwise we can set $Z = T \setminus S$ and obviously $(T \setminus Z) + i = S + i \in \mathcal{I}$. Assuming $|T \setminus S| > p$, let Z be an arbitrary set of p elements from $T \setminus S$. We consider 2 cases: If $|T \cap A| + p|T \cap B| \leq |A|$, then $|(T \setminus Z) \cap A| + p|(T \setminus Z) \cap B| \leq |A| - p$. Adding the element i can increase the left-hand side by at most p, and so $|(T \setminus Z + i) \cap A| + p|(T \setminus Z + i) \cap B| \leq |A|$. Similarly, in the second case, if $p|T \cap A| + |T \cap B| \leq |B|$, then $p|(T \setminus Z) \cap A| + |(T \setminus Z) \cap B| \leq |B| - p$. Adding the element i can increase the left-hand side by at most p, and so $p|(T \setminus Z + i) \cap A| + |(T \setminus Z + i) \cap B| \leq |B|$. ◀

Now we choose the cardinalities of A and B and the weights of their elements appropriately to get a negative result.

▶ **Lemma 12.** *For $\epsilon > 0$, let $|A| = n$ and $|B| = (p - \epsilon)n$, and set the weights as $w_a = 1$ for $a \in A$ and $w_b = p - \epsilon$ for $b \in B$. Then A is a local optimum of value $w(A) = w(B)/(p - \epsilon)^2$, unless the local search explores exchanges of size at least $\frac{\epsilon}{p}n$.*

Proof. Both A and B are independent sets. Note that for any $i \in B$, we need to remove $Z \subseteq A$ of cardinality at least $|Z| = p$ to obtain $S = (A \setminus Z) + i$ satisfying $|S \cap A| + p|S \cap B| \leq |A|$. More generally, for $Y \subseteq B$, we need to remove $Z \subseteq A, |Z| = p|Y|$ to obtain $S = (A \setminus Z) \cup Y$ that satisfies $|S \cap A| + p|S \cap B| \leq |A|$. Possibly, we could satisfy the second condition, $p|S \cap A| + |S \cap B| \leq |B|$, but this will not happen unless $|A \setminus Z| = |S \cap A| \leq |B|/p = (1 - \frac{\epsilon}{p})n$. Therefore, we would need to remove Z of cardinality at least $\frac{\epsilon}{p}n$.

If the swaps considered are smaller than $\frac{\epsilon}{p}n$ then A is a local optimum because adding $Y \subseteq B$ and removing $Z \subseteq A, |Z| = p|Y|$ results in a solution of lower weight. In conclusion, A is a local optimum of value $w(A) = n$, while the optimum is $OPT = w(B) = (p - \epsilon)^2 n$. ◀

5.3 Upper bound for p-extendible systems

Here we prove that local search does in fact provide a $1/p^2$-approximation for weighted maximization under a p-extendible system. More generally, we prove (here, we will ignore the technicalities of stopping the local search in polynomial time as this can be handled using standard techniques, while losing $1/poly(n)$ in the approximation factor) the following:

▶ **Theorem 13.** *For any p-extendible system $\mathcal{I} \subseteq 2^X$ and a monotone submodular function $f : 2^X \to \mathbb{R}_+$, local search with $(p, 1)$-swaps (including at most 1 element and removing at most p elements) provides a $1/(p^2 + 1)$-approximation. For additive f, the factor is $1/p^2$.*

Proof. Let A be a local optimum under $(p, 1)$-swaps, and let B be an optimal solution. (For convenience, let us also assume that we always try to add elements to A if possible, even if they bring zero marginal value.) We proceed in two steps, the first one inspired by the analysis of the greedy algorithm for p-extendible systems [12] and the second one similar to other analyses of local search.

Let $A = \{a_1, \ldots, a_k\}$ be a greedy ordering of A in the sense that a_1 is the element of A maximizing $f_\emptyset(a_1)$; given a_1, a_2 is the element of $A - a_1$ maximizing $f_{\{a_1\}}(a_2)$, a_3 is the element of $A - a_1 - a_2$ maximizing $f_{\{a_1,a_2\}}(a_3)$, etc. Using the p-extendible property, there is a subset $B_1 \subseteq B, |B_1| \leq p$ such that $(B \setminus B_1) + a_1 \in \mathcal{I}$. Further, since $\{a_1, a_2\} \in \mathcal{I}$, there is a subset $B_2 \subseteq B \setminus B_1, |B_2| \leq p$ such that $(B \setminus (B_1 \cup B_2)) \cup \{a_1, a_2\} \in \mathcal{I}$, etc. Generally, there are disjoint subsets $B_1, \ldots, B_k \subseteq B, |B_i| \leq p$ such that $(B \setminus (B_1 \cup \ldots B_i)) \cup \{a_1, \ldots, a_i\} \in \mathcal{I}$. In fact, if $|A| = k$, the sets $B_1, \ldots, B_k$ form a partition of B. Otherwise there would be additional elements in $B \setminus (B_1 \cup \ldots \cup B_k)$ which can be added to A, which would contradict the local optimality of A.

Now, we claim that for each $b \in B_i$, we have $f_A(b) \leq p f_{\{a_1,\ldots,a_{i-1}\}}(a_i)$. If not, we would be able to add b and, since $\{a_1, \ldots, a_{i-1}, b\} \in \mathcal{I}$, we could remove at most p elements $Z \subseteq A \setminus \{a_1, \ldots, a_{i-1}\}$ so that $(A \setminus Z) + b \in \mathcal{I}$. By submodularity and the greedy ordering, we would have $f(A \setminus Z) \geq f(A) - p f_{\{a_1,\ldots,a_{i-1}\}}(a_i)$ and again by submodularity, we would have $f((A \setminus Z) + b) \geq f(A \setminus Z) + f_A(b) > f(A \setminus Z) + p f_{\{a_1,\ldots,a_{i-1}\}}(a_i) \geq f(A)$. Therefore, this would be an improving local swap.

Since A is a local optimum, we conclude that $f_A(b) \leq p f_{\{a_1,\ldots,a_{i-1}\}}(a_i)$ for each $b \in B_i$. Since $B = B_1 \cup \ldots \cup B_k$ and $|B_i| \leq p$, we have by submodularity

$$f_A(B) \leq \sum_{i=1}^{k} \sum_{b \in B_i} f_A(b) \leq \sum_{i=1}^{k} |B_i| p f_{\{a_1,\ldots,a_{i-1}\}}(a_i) \leq p^2 \sum_{i=1}^{k} f_{\{a_1,\ldots,a_{i-1}\}}(a_i) \leq p^2 f(A)$$

For f monotone submodular, we have $f(B) \leq f(A) + f_A(B) \leq (p^2 + 1) f(A)$. For f additive, we have $f(B) = f_A(B) \leq p^2 f(A)$. This completes the proof. ◀

5.4 Recovery for p-extendible systems

▶ **Theorem 14.** *Given a p-extendible system $\mathcal{I} \subseteq 2^X$ and a monotone submodular function $f : 2^X \to \mathbb{R}_+ \cup \{0\}$ we wish to maximize, if the optimal solution B is (p^2+1)-stable, then local search with $(p, 1)$-swaps exactly recovers it. If f is additive, recovery holds if B is p^2-stable.*

Proof. The basic idea is that we can contract the elements that belong to $A \cap B$ and then use the same charging argument from above. Using the notation from the proof of Theorem 13, for elements $a_i \in A \cap B$ the corresponding B_i block is just $\{a_i\}$. Now we can rename elements in $A \setminus B = \{a_1, \ldots, a_m\}$ with corresponding blocks $B_1, \ldots, B_m$ such that $B \setminus A = B_1 \cup \ldots \cup B_m$ and $|B_i| \leq p$. Rewriting the local search guarantee:

$$f_A(B \backslash A) \leq \sum_{i=1}^{m} \sum_{b \in B_i} f_A(b) \leq \sum_{i=1}^{m} |B_i| p f_{\{a_1,\ldots,a_{i-1}\}}(a_i) \leq p^2 \sum_{i=1}^{m} f_{\{a_1,\ldots,a_{i-1}\}}(a_i) \leq p^2 f(A \backslash B)$$

Since $f_A(B \setminus A) = f(B \cup A) - f(A) \geq f(B) - f(A)$, we can $(p^2 + 1)$-perturb the input (only the marginal of elements in $A \setminus B$) and get: $\tilde{f}(B) = f(B) \leq f(A) + p^2 f(A \setminus B) = \tilde{f}(A)$, hence contradicting the $(p^2 + 1)$-stability. In the case of additive f, $f_A(B \setminus A) = f(B \setminus A)$ and $\tilde{f}(B) = f(B) = f(B \backslash A) + f(B \cap A) \leq p^2 f(A \backslash B) + f(B \cap A) \leq f(A) + (p^2 - 1) f(A \backslash B) = \tilde{f}(A)$, where we p^2-perturbed the instance, hence contradicting the p^2-stability of the instance. ◀

5.5 Recovery for the intersection of Matroids

If the independence system $\mathcal{I}$ is the intersection of p matroids: $\mathcal{I} = \cap_{i=1}^{p} \mathcal{I}_i$, local search with $(p, 1)$-swaps recovers $(p + 1)$-stable optimal solutions (for proof, see full version of the paper).

▶ **Theorem 15.** *Given $(X, \mathcal{I})$, with $\mathcal{I} = \cap_{i=1}^{p} \mathcal{I}_i$ where each $\mathcal{I}_i$ is a matroid and f monotone submodular, such that the optimal solution is $(p + 1)$-stable, Local Search exactly recovers it.*

Acknowledgements. The authors would also like to thank the anonymous reviewers for their useful comments.

References

1 Margareta Ackerman and Shai Ben-David. Clusterability: A theoretical study. In *Artificial Intelligence and Statistics*, pages 1–8, 2009.

2 Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 438–451, 2017.

3 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1):49–54, 2012.

4 Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54. ACM, 2006.

5 Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077. Society for Industrial and Applied Mathematics, 2009.

6 Maria-Florina Balcan and Mark Braverman. Nash equilibria in perturbation resilient games. *arXiv preprint arXiv:1008.1827*, 2010.

7 Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 68:1–68:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. `doi:10.4230/LIPIcs.ICALP.2016.68`.

8 Maria Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. In *International Colloquium on Automata, Languages, and Programming*, pages 63–74. Springer, 2012.

9 Shai Ben-David. Computational feasibility of clustering under clusterability assumptions. *arXiv preprint arXiv:1501.00437*, 2015.

10 Yonatan Bilu, Amit Daniely, Nati Linial, and Michael Saks. On the practically interesting instances of maxcut. *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 526–537, 2013.

11 Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(05):643–660, 2012.

12 Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

13 Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

14 Vincent Cohen-Addad, Philip N Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 353–364. IEEE, 2016.

15 Vincent Cohen-Addad and Claire Mathieu. Effectiveness of local search for geometric optimization. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 329–343, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.SOCG.2015.329`.

16 Vincent Cohen Addad and Chris Schwiegelshohn. On the local structure of stable clustering instances. *CoRR*, 2017.

17 Graham Cormode, Howard Karloff, and Anthony Wirth. Set cover algorithms for very large datasets. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 479–488. ACM, 2010.

18 Marek Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 509–518, 2013.

19 Amit Daniely, Nati Linial, and Michael Saks. Clustering is difficult only when it does not matter. *arXiv preprint arXiv:1205.4891*, 2012.

20 David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

21 Yuval Filmus and Justin Ward. The power of local search: Maximum coverage over a matroid. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 601–612. LIPIcs, 2012.

22 Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. An analysis of approximations for maximizing submodular set functions—ii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

23 Martin Fürer and Huiwen Yu. Approximating the k -set packing problem by local improvements. In *Combinatorial Optimization - Third International Symposium, ISCO 2014, Lisbon, Portugal, March 5-7, 2014, Revised Selected Papers*, pages 408–420, 2014.

24 David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

25 Bernhard Korte and Dirk Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Mathematics*, 2:65–74, 1978.

26 Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 244–257. Springer, 2009.

27 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 18–28. ACM, 2001.

28 Jan Karel Lenstra. *Local search in combinatorial optimization*. Princeton University Press, 2003.

29 László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

30 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Bilu-linial stable instances of max cut and minimum multiway cut. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'14, pages 890–906, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.

31 Julián Mestre. Greedy in approximation algorithms. In *European Symposium on Algorithms*, pages 528–539. Springer, 2006.

32 Matúš Mihalák, Marcel Schöngens, Rastislav Šrámek, and Peter Widmayer. On the complexity of the metric tsp under stability considerations. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 382–393. Springer, 2011.

33 Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 128–134. ACM, 2007.

34 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

35 Noam Nisan and Amir Ronen. Computationally feasible vcg mechanisms. *J. Artif. Intell. Res.(JAIR)*, 29:19–47, 2007.

36 Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 165–176. IEEE, 2006.

37 César Rego, Dorabela Gamboa, Fred Glover, and Colin Osterman. Traveling salesman problem heuristics: leading methods, implementations and latest advances. *European Journal of Operational Research*, 211(3):427–441, 2011.

38 Joachim Reichel and Martin Skutella. Evolutionary algorithms and matroid optimization problems. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 947–954. ACM, 2007.

39 Lev Reyzin. Data stability in clustering: A closer look. In *International Conference on Algorithmic Learning Theory*, pages 184–198. Springer, 2012.

40 Maxim Sviridenko and Justin Ward. Large neighborhood local search for the maximum set packing problem. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 792–803, 2013.

41 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.