

Revisiting the Greedy Approach to Submodular Set Function Maximization

Pranava R. Goundan

Analytics Operations Engineering, pranava@alum.mit.edu

Andreas S. Schulz

MIT Sloan School of Management, schulz@mit.edu

We consider the problem of maximizing a nondecreasing submodular set function over various constraint structures. Specifically, we explore the performance of the greedy algorithm, and a related variant, the *locally greedy* algorithm in solving submodular function maximization problems. Most classic results on the greedy algorithm and its variant assume the existence of an optimal polynomial-time *incremental oracle* that identifies, in each iteration, an element of maximum incremental value to the solution at hand. In the presence of only an approximate incremental oracle, we generalize the performance bounds of the greedy algorithm and its variant in maximizing submodular functions over (i) uniform matroids, (ii) partition matroids, and (iii) general independence systems. Subsequently, we reinterpret and, thereby, unify and improve various algorithmic results in the recent literature for problems that are specific instances of maximizing monotone submodular functions in the presence of an approximate incremental oracle. This includes results for the SEPARABLE ASSIGNMENT problem, the ADWORDS ASSIGNMENT problem, the SET k -COVER problem, basic utility games, winner determination in combinatorial auctions, and related problem variants.

Key words: approximation algorithms; greedy algorithm; matroids; submodular functions

Subject classifications: combinatorial optimization

1. Introduction

Submodular set functions are widely used in the economics, operations research, and computer science literature to represent consumer valuations, since they capture the notion of *decreasing marginal utilities* (or alternatively, *economies of scale* in a cost framework). While these properties make submodular functions a suitable candidate of choice for objective functions, submodular objective functions also arise as a natural structural form in many classic discrete optimization settings, such as the MAX SAT problem in Boolean logic, the MAX CUT problem in graphs, and the MAXIMUM COVERAGE problem in location analysis, to name a few.

The role of submodularity in discrete optimization is akin to that of convex functions in continuous optimization, given their analogous prevalence, structural properties, and the tractability of solving minimization problems on both classes of functions (Lovász (1983), Fujishige (2005)). Interestingly, submodular functions are also closely related to concave functions, and this raises the question of the tractability of maximizing submodular functions. However, since many NP-hard problems may be reduced to the problem of maximizing submodular functions, it is unlikely that there exists a polynomial-time algorithm to solve this problem. Consequently, a vast body of literature has focussed on developing efficient heuristics for various instances of this problem.

The greedy algorithm, that iteratively augments a current solution with an element of maximum incremental value, has been shown to be an effective heuristic in maximizing nondecreasing submodular functions over different constraint structures (see Conforti and Cornuéjols (1984), Farahat and Barnhart (2004), Fisher et al. (1978), Nemhauser and Wolsey (1978), Nemhauser et al. (1978), Sviridenko (2004), Wolsey (1982)). In most prior works, it was implicitly assumed that the greedy algorithm has access to an *incremental oracle* that, given a current solution, returns in polynomial time an element of highest incremental value to the current solution. However, it turns out that

in some problems, determining an element with the best incremental profit may itself be an NP-hard problem, thus necessitating the use of only an *approximate* incremental oracle. In this work, we generalize the performance bounds of the greedy algorithm and an interesting related variant, the *locally greedy* algorithm (Fisher et al. (1978)), for maximizing nondecreasing submodular set functions over various constraint structures, when the algorithm only has access to an approximate incremental oracle. Subsequently, we discuss how various results in the modern literature for problems that arise in the context of assignment problems, Internet advertising, wireless sensor networks, combinatorial auctions, and utility games, among others, may be reinterpreted and improved using these generalized performance bounds.

1.1. Preliminaries

Let E be a finite ground set. A real-valued set function $f : 2^E \rightarrow \mathbb{R}$ is **normalized**, **nondecreasing** and **submodular** if it satisfies the following conditions, respectively:

(F0) $f(\emptyset) = 0$;

(F1) $f(A) \leq f(B)$ whenever $A \subseteq B \subseteq E$;

(F2) $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq E$, or equivalently:

(F2a) $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ for all $A \subseteq B \subseteq E$ and $e \in E \setminus B$, or equivalently:

(F2b) $f(A \cup C) - f(A) \geq f(B \cup C) - f(B)$ for all $A \subseteq B \subseteq E$ and $C \subseteq E \setminus B$.

Henceforth, whenever we refer to submodular functions, we shall, in particular, imply normalized, nondecreasing, submodular functions. We also adopt the following notation: For any two sets $A, B \subseteq E$, we define the **marginal value** (incremental value) of set A to set B as

$$\rho_A(B) = f(A \cup B) - f(B).$$

Additionally, we will use the subscript e instead of $\{e\}$ whenever A is a singleton, $A = \{e\}$. In particular, **(F2a)** can equivalently be written as $\rho_e(A) \geq \rho_e(B)$ for $A \subseteq B$.

A set system (E, \mathcal{F}) , where E is a finite set and \mathcal{F} is a collection of subsets of E , is an **independence system** if it satisfies the following properties:

(M1) $\emptyset \in \mathcal{F}$;

(M2) If $X \subseteq Y \in \mathcal{F}$ then $X \in \mathcal{F}$.

Furthermore, any set $X \in \mathcal{F}$ is called an **independent** set, whereas a set $Y \in 2^E \setminus \mathcal{F}$ is called a **dependent** set. A maximal independent set in \mathcal{F} is called a **basis**.

An independence system (E, \mathcal{F}) is a **matroid** if it satisfies the additional property:

(M3) If $X, Y \in \mathcal{F}$ and $|X| > |Y|$, then there is an $x \in X \setminus Y$ with $Y \cup \{x\} \in \mathcal{F}$.

Matroids have the property that all bases in \mathcal{F} have the same cardinality.

In this work, we will also focus our attention on the following special classes of matroids:

- **Uniform matroids:** E is a finite set, k is a positive integer, and $\mathcal{F} = \{F \subseteq E : |F| \leq k\}$.

- **Partition matroids:** $E = \cup_{i=1}^k E_i$ is the disjoint union of k sets, l_1, \dots, l_k are positive integers, and

$$\mathcal{F} = \{F : F = \cup_{i=1}^k F_i \text{ where } F_i \subseteq E_i, |F_i| \leq l_i \text{ for } i = 1, \dots, k\}.$$

- **Laminar matroids** (Gabow and Kohno (2001), Calinescu et al. (2007)): Let E be a finite set. A family of subsets $\mathcal{S} \subseteq 2^E$ is said to be a laminar family if for any two sets $X, Y \in \mathcal{S}$, at least one of the three sets, $X \setminus Y$, $Y \setminus X$, $X \cap Y$ is empty. Let \mathcal{S} be a laminar family of sets, and each set $S \in \mathcal{S}$ is associated with an integer value k_S . Then,

$$\mathcal{F} = \{F \subseteq E : |F \cap S| \leq k_S \text{ for each } S \in \mathcal{S}\}.$$

Following the seminal work of Edmonds (1971), submodularity and matroids have received a lot of attention in the optimization community. The reader is referred to standard textbooks in combinatorial optimization (Cook et al. (1998), Korte and Vygen (2000), Lawler (1976), Nemhauser

and Wolsey (1988), Schrijver (2003)) for a detailed exposition on submodularity, matroids, and independence systems.

Since our focus is on developing approximation algorithms for a variety of problems, we must formalize the notion of an approximation algorithm. An α -**approximation algorithm** for a maximization problem \mathcal{P} is a polynomial-time algorithm A for \mathcal{P} such that

$$OPT(I) \leq \alpha \cdot A(I)$$

for all instances I of \mathcal{P} , where $OPT(I)$ and $A(I)$ are the optimal value and the objective value returned by the algorithm A for an instance I of \mathcal{P} . Observe that by this definition, it must be that $\alpha \geq 1$. A **fully polynomial-time approximation scheme** (FPTAS) provides, for every $\epsilon > 0$, a $(1 + \epsilon)$ -approximation algorithm whose running time is polynomial in both the size of the input and $1/\epsilon$. More generally, a **polynomial-time approximation scheme** (PTAS) provides a $(1 + \epsilon)$ approximation algorithm whose running time is polynomial in the size of the input, for any constant ϵ .

1.2. Problem Description and Literature Survey

The problem we address may be stated as follows:

$$z_{opt} = \max\{f(S) : S \subseteq E, S \in \mathcal{F}\} \quad (P)$$

where f is a normalized, nondecreasing, submodular function, and (E, \mathcal{F}) is, in general, an independence system. As stated earlier, our focus is on the performance of the greedy algorithm (and its variants), described below, in solving some special cases of this general problem.

STANDARD GREEDY ALGORITHM

Initialization: $S := \emptyset$, $E' := E$.

Incremental Oracle: Select an element $e^* \in E' \setminus S$ such that

$$e^* = \arg \max_{e \in E' \setminus S} \rho_e(S).$$

Admissibility Oracle: If $S \cup \{e^*\} \in \mathcal{F}$
Then $S := S \cup \{e^*\}$.
Else $E' := E' \setminus \{e^*\}$.

Loop back: While $E' \setminus S \neq \emptyset$ goto **Incremental Oracle**.

End

Informally stated, the greedy algorithm starts with an empty set, and in each iteration adds an element with highest marginal value to the solution using an *incremental oracle*, while ensuring independence of the resulting solution set using an *admissibility oracle* (also known as independence oracle). The algorithm continues as long as there remains an element which it has not previously considered.

A special case of the problem (P) is the maximization of a linear function over a matroid. For this problem, the greedy algorithm is known to be optimal (Rado (1957), Edmonds (1971)). Korte and Hausmann (1978) studied the problem of maximizing a linear function over an independence system and present tight bounds (that are functions of the rank quotient) on the performance of the greedy algorithm for this problem. Nemhauser et al. (1978) considered the problem (P) over a uniform

matroid and showed that greedy is a tight $(e/(e-1))$ -approximation algorithm for this problem. In a companion paper, Fisher et al. (1978) studied the problem (P) over a general independence system that is an intersection of M matroids and showed that greedy is an $(M+1)$ -approximation algorithm. This result yields a 2-approximation factor when (E, \mathcal{F}) is a matroid. The authors also considered a simpler variant of the greedy algorithm, that they refer to as the *locally greedy heuristic*, and showed that this algorithm is also a factor-2 approximation algorithm for the problem (P) when (E, \mathcal{F}) is a partition matroid. Subsequently, Conforti and Cornuéjols (1984) studied the problem (P) over a matroid, but for a richer class of objective functions, f , by introducing the notion of *total curvature* to characterize a set function. They showed that the performance of the greedy algorithm for maximizing a nondecreasing submodular set function of total curvature α is an $(\alpha+1)$ -approximation. Moreover, by showing that $0 \leq \alpha \leq 1$ for nondecreasing submodular functions and $\alpha = 0$ if and only if the function is linear, they generalized the results of Rado-Edmonds and Fisher et al. (1978) regarding the performance of the greedy algorithm.

Wolsey (1982) considered the problem (P) over an independence system (E, \mathcal{F}) given by:

$$\mathcal{F} = \{S \subseteq E : \sum_{e \in S} w_e \leq W\}$$

where w_e , for each $e \in E$, are nonnegative weights and W is a nonnegative integer. This system is simply the set of all feasible solutions to a knapsack constraint, and exemplifies independence systems where \mathcal{F} may be exponentially large, and yet may be encoded succinctly in a problem instance. In what follows, we will see examples where the ground set E itself may be exponentially large and yet may be encoded concisely in a problem instance. Extending a result of Khuller et al. (1999) regarding the performance of a *greedy with partial enumeration* algorithm for the BUDGETED MAXIMUM COVERAGE problem, Sviridenko (2004) showed that this algorithm is also an $(e/(e-1))$ -approximation algorithm for the problem (P) over a knapsack independence system. The $(e/(e-1))$ -approximation results of Sviridenko (2004) and Nemhauser et al. (1978) for their respective problems are in fact *best possible* for any polynomial-time approach, unless $P=NP$ (Feige (1998)).

Upon the completion of this work, we learnt that recently Calinescu et al. (2007) have developed a pipage rounding based $(e/(e-1))$ -approximation algorithm for the case of problem (P) where (E, \mathcal{F}) is a matroid and f is a sum of weighted rank functions of matroids, which are a rich subclass of monotone submodular functions. Moreover, the authors also give a somewhat different proof for the performance of the standard greedy algorithm with an approximate oracle for the problem (P) when (E, \mathcal{F}) is a p -independent family.

2. Motivation

Whereas all of the works highlighted in the previous section, with the exception of the recent paper of Calinescu et al. (2007), assume the existence of a polynomial-time procedure (or incremental oracle) in the greedy algorithm to find an optimal incremental element in each iteration, such an oracle may not always be available. We motivate this scenario via an example where the ground set, E , itself may be exponentially large. Consider the following problem studied by Fleischer et al. (2006):

SEPARABLE ASSIGNMENT

Instance: A set, U , of n items and a set, B , of m bins. Each bin $i \in B$ has an independence system \mathcal{I}_i of subsets of items that fit in bin i . A profit p_{ij} for assigning item j to bin i .

Task: Find a subset of items, $S \subseteq U$, and an assignment of these items, $S_i \in \mathcal{I}_i$ to bin i , $S_i \cap S_k = \emptyset$ for $i \neq k$, so as to maximize profit, $\sum_{i \in B} \sum_{j \in S_i} p_{ij}$.

Observe that the family of feasible subsets for each bin i , \mathcal{I}_i , is an independence system. Also note that therefore, the constraints defining feasible packings for bin i , implicit in \mathcal{I}_i , are *separable* from the constraints for bin j , i.e., the set of feasible packings of bin i are unaffected by the set of feasible packings of bin j . Finally, the authors assume the existence of an α -approximation algorithm for the single-bin subproblem for each bin i : select a feasible packing of items from \mathcal{I}_i of maximum profit. As an example, the GENERALIZED ASSIGNMENT problem is a special case of the SEPARABLE ASSIGNMENT problem, where the single bin subproblem is the KNAPSACK problem. Specifically, in the GENERALIZED ASSIGNMENT problem, items also have sizes w_{ij} corresponding to each bin i , and each bin itself is a knapsack of a particular capacity B_i . Hence, the single-bin subproblem corresponding to bin i for GENERALIZED ASSIGNMENT would be to find a maximum profit subset of items that fits in bin i . However, for other special cases of SEPARABLE ASSIGNMENT, the single-bin subproblem may be characterized by other forms of resource packing problems, such as the RECTANGLE PACKING or the 2-DIMENSIONAL KNAPSACK problem.

As noted independently by Chekuri (2006) and by Fleischer et al. (2006), this problem is an instance of maximizing a normalized, nondecreasing, submodular function over a (partition) matroid. For the sake of completeness, and to illustrate a case for which the ground set E of the matroid is exponentially large and finding the best incremental element may be NP-hard, we describe this transformation here.

OBSERVATION 1. SEPARABLE ASSIGNMENT is an instance of maximizing a monotone submodular function over a partition matroid.

Proof. For any instance of the SEPARABLE ASSIGNMENT problem, define a ground set $E = \cup_{i \in B} E_i$, with an element $e_S \in E_i$ corresponding to each feasible packing, $S \in \mathcal{I}_i$, of bin i . The constraints on SEPARABLE ASSIGNMENT now transform to picking at most one element from each set E_i . Let $F = \cup_{i \in B} F_i$, where $F_i \subseteq E_i$ and $|F_i| \leq 1$, represent the set of elements picked. This underlying constraint structure is therefore a partition matroid. However, note that the packings of bins corresponding to any set F , may contain multiple copies of the same item. Therefore it is important that one does not double-count the profit for these items. This may be taken care of by writing the objective function as:

$$f(F) = \sum_{j \in U} \max\{p_{ij} : i \in B, e_S \in F_i, j \in S\} .$$

Observe that this definition of f extends to all subsets $F \subseteq E$, even when $|F_i|$ is more than 1. Observe also that the summation is over all items $j \in U$, and the maximum is over all bins i that contain item j . In other words, if an item j is in multiple bins corresponding to a set, F , then out of all the bins, i , that item j is in, we assign only the maximum p_{ij} value to item j . It is not hard to verify that indeed this function f is nondecreasing, and has decreasing marginal values. Suppose that $S \in \mathcal{I}_i$ is a feasible packing for bin i . Observe that the incremental value of element e_S to a set F , $\rho_{e_S}(F)$, is given by:

$$\rho_{e_S}(F) = f(F \cup \{e_S\}) - f(F) = \sum_{j \in S} \max\{p_{ij} - \max\{p_{kj} : e_P \in F, P \in \mathcal{I}_k, j \in P\}, 0\}.$$

Intuitively, the incremental value of an element is the incremental profit value of the set of items in the corresponding packing that the element represents. As the set F grows, the likelihood of the items in the packing, S , of bin i having the highest profit, p_{ij} , decreases and hence, f has decreasing marginal values. Therefore, f is submodular. \square

We have seen that in the underlying matroid, an element of the ground set corresponds to a feasible packing in a bin. Consequently, the role of an incremental oracle in the greedy algorithm for this problem is to pick a feasible packing among all feasible packings of maximum incremental value to the existing solution. This would typically involve solving a knapsack problem (or even a rectangle packing problem), since the set of all feasible packings might be exponentially large. However, since such packing problems are typically NP-hard, we cannot hope to have an optimal incremental oracle, unless P=NP. Hence, generalized results such as the one described below, assuming instead the existence of an α -approximation oracle to find a “good” incremental element, are in order.

In this work, we present bounds on the performance of a greedy algorithm that uses an α -approximation algorithm as the incremental oracle to determine an incremental element to add to the greedy solution. We begin by considering the problem of maximizing a nondecreasing submodular function over uniform matroids. Generalizing a previous result due to Nemhauser et al. (1978), we show that in the presence of an α -approximate incremental oracle, the standard greedy algorithm is an $(e^{1/\alpha}/(e^{1/\alpha} - 1))$ -approximation algorithm for this problem. Further, we also discuss how our result generalizes similar previous results due to Hochbaum and Pathria (1998) in the context of the MAXIMUM COVERAGE problem, and Chekuri and Khanna (2006) with regards to the MULTIPLE KNAPSACK problem with identical bin capacities.

Partition matroids generalize uniform matroids, in that the ground set E contains elements of different kinds, with individual restrictions on how many elements may be selected of each kind. In Section 4, we consider a variant of the standard greedy algorithm, namely the locally greedy algorithm, previously proposed by Fisher et al. (1978) and consider the performance of this algorithm for maximizing nondecreasing submodular set functions over partition matroids. Extending a result of Fisher et al. (1978) to α -approximate incremental oracles, we show that the locally greedy algorithm guarantees a tight factor- $(\alpha + 1)$ result for the submodular function maximization problem over partition matroids. We also show that various optimization problems that arise in the context of the winner determination in combinatorial auctions (Lehmann et al. (2006)), generalized assignment problems (Fleischer et al. (2006)), AdWords assignment problems (Fleischer et al. (2006)), basic utility games (Vetta (2002), Mirrokni and Vetta (2004)), wireless networks (Abrams et al. (2004)), etc. may be cast into the framework of maximizing a submodular function over a partition matroid. Consequently, we *reinterpret, unify, and even improve* upon some of the results pertaining to these problems.

In Section 5, we consider the problem of maximizing a submodular function over an independence system. If the independence system is an intersection of a finite number, M , of matroids, then Fisher et al. (1978) showed that the greedy algorithm with an optimal incremental oracle is an $(M + 1)$ -approximation algorithm for this problem. When only an α -approximate incremental oracle is available, we show that the greedy algorithm is an $(\alpha M + 1)$ -approximation for the problem. Based on this result, we improve upon a previous result of Fleischer et al. (2006) for the k -MEDIAN WITH HARD CAPACITIES AND PACKING problem and we present a greedy $(\alpha + 1)$ -approximation algorithm for it. Finally, we conclude in Section 6 by highlighting some interesting open questions and future directions that result from this work.

3. Generalized Results over Uniform Matroids

Let (E, \mathcal{F}) be a uniform matroid, i.e., $\mathcal{F} = \{S \subseteq E : |S| \leq k\}$ for some integer k . Consider the problem of maximizing a normalized, nondecreasing, submodular function f over this uniform

matroid. Using notation introduced by Farahat and Barnhart (2004), we represent this problem as $f_S|\mathcal{F}_U$. We describe a generalized greedy algorithm below that uses an α -approximation algorithm as the incremental oracle to find an element e with the best incremental value, $\rho_e(S) = f(S \cup \{e\}) - f(S)$. Note that in the case of uniform matroids, the role of the admissibility oracle in the greedy algorithm is trivial – as long as the size of the solution set, S , is strictly smaller than k , any element is admissible.

GREEDY ALGORITHM FOR $f_S|\mathcal{F}_U$

Step 1: Set $i := 1$; let $S_0 := \emptyset$.

Step 2: Select an element $e_i \in E \setminus S_{i-1}$ for which $\alpha \rho_{e_i}(S_{i-1}) \geq \max_{e \in E \setminus S_{i-1}} \rho_e(S_{i-1})$ using an α -approximate incremental oracle.

Step 3: Set $S_i := S_{i-1} \cup \{e_i\}$.

Step 4: Set $i := i + 1$. If $i \leq k$, then goto **Step 2**.

We use S_i to represent the set generated by the greedy algorithm after i iterations. Let $S^G = S_k$ be the solution returned by the greedy algorithm. Let ρ_i represent the incremental profit obtained by the addition of element e_i to the set S_{i-1} . Let ρ'_i represent the optimal incremental profit that could have been obtained, given the set S_{i-1} was selected by the first $i - 1$ iterations of the greedy algorithm. Since we use an α -approximation oracle in order to determine the element with the best incremental objective function value, it follows that $\rho_i \leq \rho'_i \leq \alpha \rho_i$. Observe that if one had access to an optimal incremental oracle, then it would be the case that $\rho_i \geq \rho_{i+1}$. However, since we only use an approximate incremental oracle, this need not hold anymore. Thus, *the use of an approximate incremental oracle does not preserve the nonincreasing property of incremental values of elements selected by the greedy algorithm*. We begin by noting the following characterization for nondecreasing submodular functions:

LEMMA 1 (Nemhauser et al. (1978)). *f is a nondecreasing submodular set function on E if and only if $f(T) \leq f(S) + \sum_{j \in T \setminus S} \rho_j(S)$ for all $S, T \subseteq E$.*

Suppose that $z_{opt} = \max_{S \subseteq E} \{f(S) : |S| \leq k\}$, with f is normalized, nondecreasing, and submodular. We then show that:

THEOREM 1. *If z_g is the value of the GREEDY ALGORITHM FOR $f_S|\mathcal{F}_U$, then $\frac{z_{opt}}{z_g} \leq \frac{(\alpha k)^k}{(\alpha k)^k - (\alpha k - 1)^k} \leq \frac{e^{1/\alpha}}{e^{1/\alpha} - 1}$.*

Proof. Suppose that S^G is the set generated by the greedy algorithm and T is an optimal solution to the above problem. Let ρ'_i represent the best incremental value that could have been obtained during the i^{th} iteration of the greedy algorithm. By substituting $S = \emptyset$ in Lemma 1 and observing that $|T| \leq k$, it follows that:

$$z_{opt} = f(T) \leq \sum_{j \in T} f(j) \leq k \rho'_1 \leq k(\alpha \rho_1).$$

Now, applying Lemma 1 to the solution of the greedy algorithm after j iterations, S_j , implies that:

$$z_{opt} \leq f(S_j) + \sum_{i \in T \setminus S_j} \rho_i(S_j). \quad (1)$$

Given that $f(S_j) = \sum_{i=1}^j \rho_i$ and that

$$\alpha \rho_{j+1} \geq \rho'_{j+1} \geq \rho_i(S_j) \text{ for all } i \in E \setminus S_j,$$

equation (1) now yields the following inequality:

$$z_{opt} \leq \sum_{i=1}^j \rho_i + k (\alpha \rho_{j+1}),$$

which implies that

$$\rho_{j+1} \geq \frac{1}{\alpha k} z_{opt} - \frac{1}{\alpha k} \sum_{i=1}^j \rho_i.$$

Adding $\sum_{i=1}^j \rho_i$ on both sides of the above inequality, we get an inequality of the form:

$$\sum_{i=1}^{j+1} \rho_i \geq \frac{1}{\alpha k} z_{opt} + \frac{(\alpha k - 1)}{\alpha k} \sum_{i=1}^j \rho_i. \quad (2)$$

We now prove by induction on j that:

$$\sum_{i=1}^j \rho_i \geq \frac{(\alpha k)^j - (\alpha k - 1)^j}{(\alpha k)^j} z_{opt}.$$

For $j = 1$, we have that $\rho_1 \geq \frac{1}{\alpha k} z_{opt}$. Assume that the claim holds for $j - 1$. Now, applying the induction hypothesis on equation (2), we have:

$$\sum_{i=1}^j \rho_i \geq \frac{1}{\alpha k} z_{opt} + \frac{\alpha k - 1}{\alpha k} \frac{(\alpha k)^{j-1} - (\alpha k - 1)^{j-1}}{(\alpha k)^{j-1}} z_{opt}.$$

Simplifying the right-hand side of the above expression yields the induction claim. Finally, setting $j = k$ we have:

$$z_g = \sum_{i=1}^k \rho_i \geq \frac{(\alpha k)^k - (\alpha k - 1)^k}{(\alpha k)^k} z_{opt},$$

which proves the approximation ratio claim that:

$$\frac{z_{opt}}{z_g} \leq \frac{(\alpha k)^k}{(\alpha k)^k - (\alpha k - 1)^k} \leq \frac{e^{1/\alpha}}{e^{1/\alpha} - 1}. \quad \square$$

The above result essentially follows in a manner similar to that of Nemhauser et al. (1978) and serves to point out the effect of α on the approximation factor of the greedy algorithm. For the case when $\alpha = 1$, the result is precisely that of Nemhauser et al. (1978) and therefore tight. Theorem 1 also generalizes a similar result due to Hochbaum and Pathria (1998) (see also Hochbaum (1997)) in the context of the MAXIMUM COVERAGE problem and its applications. We discuss this in greater detail in Section 3.2 below.

3.1. Discussion on Running Time of Greedy Algorithm

Denote the running time of the α -approximate incremental oracle by P . It follows then that the running time of the GREEDY ALGORITHM FOR $f_S|\mathcal{F}_U$ is $O(kP)$, where at most k elements need to be selected from the uniform matroid. Observe that the running time of the algorithm itself does not depend on the size of the ground set, E , which may possibly be exponentially large. As discussed earlier, and motivated by the SEPARABLE ASSIGNMENT problem, in certain underlying problems, the ground set E may be encoded concisely, even though it is exponentially large. In that sense, the greedy algorithm is an efficient algorithm as long as P is polynomial in the input size of the underlying problem. Moreover, for problems such as the relaxed AD PLACEMENT problem, where the number of elements to be selected is itself encoded using $\log k$ bits, it is often possible to modify the greedy algorithm appropriately so that its running time is still polynomial, as demonstrated by Goundan and Schulz (2007b) for the relaxed AD PLACEMENT problem.

3.2. Applications of Generalized Result

We begin by studying implications of Theorem 1 for the MAXIMUM COVERAGE problem in discrete optimization. The MAXIMUM COVERAGE problem may be stated as follows:

MAXIMUM COVERAGE

Instance: A set of elements, U , a collection \mathcal{R} of subsets of U , and an integer k . A nonnegative profit, p_j , corresponding to each element $j \in U$.

Task: Select k subsets U_1, \dots, U_k of U , with each $U_i \in \mathcal{R}$, such that the profit of the elements in $\cup_{i=1}^k U_i$ is maximized.

Vohra and Hall (1993) noted that MAXIMUM COVERAGE is indeed a special case of $f_S|\mathcal{F}_U$. Hochbaum and Pathria (1998) presented a greedy algorithm to solve MAXIMUM COVERAGE, and a scenario where finding a subset that gives maximum improvement might be hard. They obtained the same bound as the one in Theorem 1, assuming that one is able to pick an α -approximate solution in each stage. Hochbaum and Pathria (1998) also described a number of applications that can be modeled as the MAXIMUM COVERAGE problem in a setting of approximate improvement.

Theorem 1 also implies the bound on the performance of the greedy algorithm obtained by Chekuri and Khanna (2006) for the MULTIPLE KNAPSACK problem with identical bin capacities. The MULTIPLE KNAPSACK problem may be stated as follows:

MULTIPLE KNAPSACK

Instance: Nonnegative integers, $n, m, p_1, \dots, p_n, w_1, \dots, w_n$, and W_1, \dots, W_m .

Task: Find m subsets $S_1, \dots, S_m \subseteq \{1, \dots, n\}$, $S_i \cap S_k = \emptyset$ for $i \neq k$, such that $\sum_{j \in S_i} w_j \leq W_i$ for $i = 1, \dots, m$ and $\sum_{i=1}^m \sum_{j \in S_i} p_j$ is maximum.

In the case that all m bins have the same capacity, $W_1 = W_2 = \dots = W_m = W$, this problem is an instance of $f_S|\mathcal{F}_U$. The transformation for this is essentially identical to that described for SEPARABLE ASSIGNMENT in Observation 1. Using an FPTAS for the KNAPSACK problem as an incremental oracle, the greedy $(\frac{e}{e-1} + \epsilon)$ -approximation result of Chekuri and Khanna (2006) follows from Theorem 1.

4. The Locally Greedy Algorithm and Partition Matroids

In this section, we generalize the performance bounds of a special version of the greedy algorithm, namely the *locally greedy heuristic* of Fisher et al. (1978), to maximize a submodular function over a partition matroid. Recall that a partition matroid, (E, \mathcal{F}) is given by $\mathcal{F} = \{F : F = \cup_{i=1}^k F_i \text{ where } F_i \subseteq E_i, |F_i| \leq l_i \text{ for } i = 1, \dots, k\}$. We assume that we only have at our disposal an α -approximation algorithm to play the role of an incremental oracle for each element type, E_i . We shall refer to this problem as $f_S|\mathcal{F}_P$, where the subscript P denotes the partition matroid. The locally greedy algorithm for this problem is as follows:

LOCALLY GREEDY ALGORITHM FOR $f_S|\mathcal{F}_P$

Step 1: Set $i := 1$; let $S_0 := \emptyset$; $m := 1$.

Step 2: Set $j := 1$.

Step 3: Select an element $e_j \in E_i \setminus S_{m-1}$ for which $\alpha \rho_{e_j}(S_{m-1}) \geq \max_{e \in E_i \setminus S_{m-1}} \rho_e(S_{m-1})$ using an α -approximation algorithm as the incremental oracle of type i .

Step 4: Set $S_m := S_{m-1} \cup \{e_j\}$; $m := m + 1$.

Step 5: Set $j := j + 1$. If $j \leq l_i$, then goto Step 3.

Step 6: Set $i := i + 1$. If $i \leq k$, then goto Step 2.

In the above algorithm, i is a counter of the type, E_i , of elements in consideration; j is a counter for the number of elements selected within a particular type; and m represents the number of elements selected by the greedy algorithm at any point. The locally greedy algorithm basically begins by selecting “profitable” elements of the first type, E_1 , from E , until it has picked l_1 elements from E_1 , and then it proceeds to do so for the second type of elements, E_2 in E and so on. Thus, the number of elements, m , in the greedy solution, S^G , is at most $\sum_{i=1}^k l_i$.

It is important to note that, *the order in which the locally greedy algorithm deals with elements of different types is completely arbitrary*. Furthermore, the incremental oracle in the locally greedy algorithm only need select *an approximate best element within each particular type*, rather than an approximate best element across all element types. Of course, with an α -approximate incremental oracle over each type, one may simulate an α -approximate incremental oracle over all types in $O(k)$ time, by taking the α -best element of each of k types and selecting the best of them.

Our main result for the maximization of submodular set functions over partition matroids is that:

THEOREM 2. *If z_g is the value of the solution provided by the LOCALLY GREEDY ALGORITHM FOR $f_S|\mathcal{F}_P$, and z_{opt} is the value of an optimal solution, then $\frac{z_{opt}}{z_g} \leq \alpha + 1$.*

Proof. Let S^G represent the greedy solution and T an optimal solution to $f_S|\mathcal{F}_P$. Substituting them in Lemma 1, we have that:

$$z_{opt} = f(T) \leq f(S^G) + \sum_{j \in T \setminus S^G} \rho_j(S^G).$$

Now, suppose $T \setminus S^G = \cup_{i=1}^k T_i$, where $T_i \subseteq E_i$. Also, suppose $S^G = \cup_{i=1}^k S_i^G$ where $S_i^G \subseteq E_i$. Let e_i be the element in S_i^G that was selected with the lowest ρ value, which was at a point in the algorithm when the current greedy solution, just before the addition of e_i , was S^{e_i} . Mathematically,

$$\rho_{e_i}(S^{e_i}) = \min_{e \in S_i^G} \rho_e(S^{e_i}).$$

In other words, $\rho_{e_i}(S^{e_i})$ is the minimum incremental value selected by the greedy algorithm among elements in S_i^G . But given that we are using an α -approximation algorithm, we also know that:

$$\alpha \rho_{e_i}(S^{e_i}) \geq \rho_e(S^{e_i}) \quad \text{for all } e \in E_i \setminus S^{e_i}. \quad (3)$$

Also, since $\sum_{j \in T \setminus S^G} \rho_j(S^G) = \sum_{i=1}^k \sum_{j \in T_i} \rho_j(S^G)$, it now follows that:

$$\begin{aligned} z_{opt} &\leq f(S^G) + \sum_{i=1}^k \sum_{j \in T_i} \rho_j(S^G) \\ &\leq z_g + \sum_{i=1}^k \sum_{j \in T_i} \rho_j(S^{e_i}). \end{aligned} \quad (4)$$

Inequality (4) follows from the submodularity of f (see property **(F2a)**), since $S^{e_i} \subseteq S^G$, for $i = 1, \dots, k$. Noting that $T_i \subseteq E_i \setminus S^G \subseteq E_i \setminus S^{e_i}$ and using inequality (3), the right-hand side of inequality (4) yields further that:

$$\begin{aligned} z_{opt} &\leq z_g + \sum_{i=1}^k \sum_{j \in T_i} \alpha \rho_{e_i}(S^{e_i}) = z_g + \sum_{i=1}^k \alpha |T_i| \rho_{e_i}(S^{e_i}) \\ &\leq z_g + \alpha z_g \end{aligned} \quad (5)$$

Inequality (5) is implied from the way we picked e_i to be the element with the lowest incremental function value in S_i^G , and since we may assume without loss of generality that $|T_i| \leq |S_i^G| \leq l_i$ (recall that (E, \mathcal{F}) is a matroid and f is nondecreasing, and therefore we can always add elements to S_i^G so that $|T_i| \leq |S_i^G|$). \square

If one uses an optimal incremental oracle in the greedy algorithm, implying $\alpha = 1$, then Theorem 2 matches the result of Fisher et al. (1978), and guarantees a bound of 2 for the performance of a locally greedy algorithm over $f_S|_{\mathcal{F}_P}$. This result of Fisher et al. (1978) for the locally greedy heuristic seems relatively unknown compared to their result for the performance of the standard greedy algorithm for $f_S|_{\mathcal{F}_M}$, where the subscript M denotes arbitrary matroids. In Section 4.2, we reinterpret a few results in the literature based on the result of Fisher et al. (1978) for the locally greedy heuristic. But first, we quickly comment on the running time of the locally greedy algorithm.

4.1. Discussion on Running Time of Locally Greedy Algorithm

It is easy to see that the running time of the LOCALLY GREEDY ALGORITHM FOR $f_S|_{\mathcal{F}_P}$ depends only on the number of elements, l_i , to be picked of each type i , and the running time of the incremental oracle of type i , P_i , as $O(\sum_{i=1}^k l_i P_i)$. It does not depend on the size of the ground set, E , which may potentially be exponentially large, as in SEPARABLE ASSIGNMENT. It follows that the locally greedy algorithm is an efficient algorithm as long as the running time, P_i , of each oracle is polynomial in the size of the input of the underlying optimization problem.

4.2. Applications of Generalized Result for Partition Matroids

We now relate some recent work in the literature to the result of Fisher et al. (1978) for the locally greedy heuristic:

Winner determination problem in combinatorial auctions. When all buyers in a combinatorial auction possess nondecreasing submodular valuations, it has been observed that the WINNER DETERMINATION problem is a special case of $f_S|\mathcal{F}_P$ (see Appendix A for details). Interestingly, we point out that the factor-2 greedy approximation algorithm proposed by Lehmann et al. (2006) for this problem turns out to be exactly the locally greedy heuristic of Fisher et al. (1978), as both algorithms are independent of item ordering. To complete the analogy between both these greedy algorithms, we observe that Lehmann et al. (2006) assume access to a *value oracle* to encode the submodular valuations of the players. That is, given a set S , the value oracle outputs $f(S)$. In the corresponding locally greedy algorithm, the existence of an optimal incremental oracle follows from the existence of the value oracle, and from the fact that there are only n buyers whose valuations need to be checked to find the best incremental element.

Convergence issues in competitive games. We can show that basic-utility games with nondecreasing submodular social utility functions, as defined by Vetta (2002), are an instance of $f_S|\mathcal{F}_P$, where the underlying elements of type i in the partition matroid simply correspond to elements in the player i 's strategy space. Furthermore, a one-round best-response path of this game starting from \emptyset , as introduced by Mirrokni and Vetta (2004), corresponds to the execution of the locally greedy algorithm. The factor-2 result of Mirrokni and Vetta (2004) for one-round paths follows as a consequence. For further details, we refer the reader to Appendix B.

Set k -Cover problems in wireless sensor networks. Abrams et al. (2004) studied a variant of the SET k -COVER problem that we show is an instance of $f_S|\mathcal{F}_P$ (see Appendix C). They also developed a distributed greedy algorithm that is a 2-approximation algorithm for the problem. This distributed greedy algorithm is in fact analogous to the locally greedy algorithm.

Based on the result of Theorem 2, we now put into perspective other results in the literature where in the absence of an optimal incremental oracle, the locally greedy algorithm uses an α -approximate incremental oracle. As mentioned earlier, the SEPARABLE ASSIGNMENT problem is an instance of $f_S|\mathcal{F}_P$. Fleischer et al. (2006) devised a polynomial-time local search $(\alpha + 1 + \epsilon)$ -approximation algorithm for SEPARABLE ASSIGNMENT, given an α -approximation algorithm for the single-bin subproblem. It may be seen that any such α -approximation algorithm for the single-bin subproblem corresponds exactly to an α -approximate incremental oracle for the locally greedy algorithm. Theorem 2 therefore implies that:

COROLLARY 1. *There is a polynomial-time locally greedy $(\alpha + 1)$ -approximation algorithm for SEPARABLE ASSIGNMENT, given an α -approximation algorithm for the single-bin subproblem.*

Fleischer et al. (2006) proposed a linear programming-based $\alpha e/(e - 1)$ -approximation algorithm for SEPARABLE ASSIGNMENT, given an α -approximation algorithm for the single-bin subproblem. However, observe that if $\alpha \geq (e - 1)$, then $(\alpha + 1) \leq \alpha e/(e - 1)$. Hence, if we only have “weak” approximation algorithms for the single-bin subproblem (such as in the RECTANGLE PACKING problem), the locally greedy algorithm outperforms the LP-based algorithm for SEPARABLE ASSIGNMENT.

Chekuri and Khanna (2006) proved that for the MULTIPLE KNAPSACK problem, the performance ratio of a greedy algorithm solving the KNAPSACK problem successively is $(2 + \epsilon)$, and noted that the same result holds even when the weights of items vary across bins. Also, Dawande et al. (2000) proposed a similar greedy algorithm-based $(2 + \epsilon)$ -result for a MULTIPLE KNAPSACK problem with “assignment restrictions,” wherein items are restricted to be assigned only to certain specified sets of bins. We note that both these results follow from Theorem 2 as well, since these problems are special cases of SEPARABLE ASSIGNMENT.

Chekuri and Kumar (2004) studied a variant of the MAXIMUM COVERAGE problem, that they called MAXIMUM COVERAGE WITH GROUP BUDGET CONSTRAINTS. They also considered the

performance of a greedy algorithm that uses an α -approximate incremental oracle and showed that the performance of their greedy algorithm for the cardinality version of MAXIMUM COVERAGE WITH GROUP BUDGET CONSTRAINTS is $(\alpha + 1)$. By observing that the cardinality version of MAXIMUM COVERAGE WITH GROUP BUDGET CONSTRAINTS is a special case of $f_S|\mathcal{F}_P$, and that their greedy algorithm is analogous to the locally greedy algorithm, Theorem 2 implies the same result.

The AdWords Assignment Problem. Search-based advertising is, increasingly, the most popular form of advertising on the Internet, and a significant source of revenue for search portals such as Google, Yahoo, and MSN, to name a few. In this paradigm, portals solicit advertisements for particular *keywords*, called “AdWords” in the case of Google. The phrase, “linear programming,” would be an example of one such keyword, with which advertisers might like to associate their advertisement. When a user of a search portal types in a *query*, it is matched with a corresponding keyword and the associated advertisements are displayed on the search results page. Hence, advertisers would have different *valuations* for their advertisement being associated with different keywords, which in turn, would depend on which queries match each keyword.

If a search firm is aware of advertisers’ private valuations and, therefore, their willingness to pay for each keyword, then one might think of framing the search firm’s problem of assigning advertisements to keywords to maximize revenue as the following optimization problem proposed by Fleischer et al. (2006):

ADWORDS ASSIGNMENT

Instance: A set of n bidders with a budget B_i for each bidder i ; a rectangular ad A_i of length l_i and width w_i that bidder i would like to advertise; a set of m AdWords (keywords), each AdWord j with a rectangular display area of length L_j and width W_j ; bidder i has a maximum willingness to pay, v_{ij} , for having its ad associated with AdWord j .

Task: Find a feasible assignment of AdWords, S_i , to bidder i so as to maximize total revenue, where the revenue obtained from bidder i is $\min(B_i, \sum_{j \in S_i} v_{ij})$. In a feasible assignment, all ads assigned to AdWord j must be feasibly displayed in the rectangular display without having to rotate any of the ads.

It turns out this ADWORDS ASSIGNMENT is in fact a special case of maximizing a nondecreasing submodular function over a partition matroid. We present this transformation below.

LEMMA 2. *The ADWORDS ASSIGNMENT problem is a special case of maximizing a nondecreasing submodular function over a partition matroid.*

Proof. Consider an underlying ground set $E = \cup_{j=1}^m E_j$, where each element $e_S \in E_j$ corresponds to a feasible assignment of ads, S , that may be accommodated in the rectangular display corresponding to AdWord j . A feasible solution to the ADWORDS ASSIGNMENT problem would constrain that at most one element may be picked of each type, E_j , thereby defining a partition matroid on E .

For any subset, $F \subseteq E$, suppose that $F_j = F \cap E_j$. Furthermore, let $F_j^i = \{e_S \in F_j : \text{ad } i \in S\}$. Define a function, f , on a subset $F \subseteq E$ as follows:

$$f(F) = \sum_{i=1}^n \min \left(B_i, \sum_{j=1}^m v_{ij} |F_j^i| \right) .$$

It is not hard to verify that f is exactly the objective function of ADWORDS ASSIGNMENT over all feasible sets in the partition matroid, and therefore over all feasible solutions of ADWORDS

ASSIGNMENT. Moreover, observe that f is indeed nondecreasing and has decreasing marginal values, by construction. Hence the claim holds. \square

From the transformation presented above, it is not hard to see that the role of the incremental oracle in a locally greedy algorithm for this problem would be played by an algorithm for the RECTANGLE PACKING problem:

RECTANGLE PACKING

Instance: Set of n rectangles, $R_i = (l_i, w_i, p_i)$, where $l_i \leq L$ and $w_i \leq W$ are the length and width of R_i , respectively, and p_i is the profit associated with R_i . Also, a big rectangle R of length L and width W .

Task: Find a subset of rectangles $S \subseteq \{R_1, \dots, R_n\}$ that can be feasibly packed in R which maximizes $\sum_{i \in S} p_i$.

For RECTANGLE PACKING, the best-known result is a $(2 + \epsilon)$ -approximation scheme due to Jansen and Zhang (2004). Consequently, Theorem 2 implies that:

COROLLARY 2. *The locally greedy algorithm, with a $(2 + \epsilon)$ -approximation scheme for RECTANGLE PACKING as an approximate incremental oracle, is a $(3 + \epsilon)$ -approximation scheme for the ADWORDS ASSIGNMENT problem.*

The above result improves on the previous best $(2 + \epsilon)e/(e - 1) \approx (3.16 + \epsilon)$ -approximation result of Fleischer et al. (2006) for the ADWORDS ASSIGNMENT problem.

It is instructive to understand the difference between the locally greedy algorithm and the standard greedy algorithm. A standard greedy algorithm in any iteration tries to pick the “best” incremental element in E over all element types, and does not constrain itself to pick only from a certain subset E_i . One might therefore expect that for partition matroids, the standard greedy performs better than the locally greedy algorithm. However, as it turns out, even the standard greedy algorithm achieves the same approximation factor as the locally greedy algorithm, and this factor is tight.

OBSERVATION 2. For the problem $f_S | \mathcal{F}_P$, the worst-case performance of a standard greedy algorithm as well as a locally greedy algorithm using an α -approximate incremental oracle is no better than $(\alpha + 1)$.

Proof. Consider a partition matroid, with $E = E_1 \cup E_2$, with $E_1 = \{a, b\}$, $E_2 = \{c\}$, where at most one element may be picked from E_1 and E_2 , respectively, and a submodular function, f , defined as $f(\emptyset) = 0$, $f(\{a\}) = \alpha$, $f(\{b\}) = f(\{c\}) = 1$, $f(\{a, c\}) = f(\{a, b\}) = \alpha + 1$, $f(\{b, c\}) = 1$, $f(\{a, b, c\}) = \alpha + 1$. It may be easily verified that f is indeed normalized, nondecreasing and submodular. Moreover, the optimal solution in this instance yields a value of $f(\{a, c\}) = \alpha + 1$. However, the standard greedy algorithm and the locally greedy algorithm may yield the solution $f(\{b, c\}) = 1$, by picking b in the first iteration, using an α -approximate incremental oracle. Thus the approximation guarantee of both these algorithms is $(\alpha + 1)$. \square

We end this section by pointing out that indeed, for a lot of the problems discussed, there are algorithms with better performance guarantees than that of the locally greedy algorithm. For example, Fleischer et al. (2006) gave an $(e/(e - 1))$ -approximation algorithm for the GENERALIZED ASSIGNMENT problem, that has recently been improved to $(e/(e - 1) - \epsilon)$ by Feige and Vondrák (2006). For the MULTIPLE KNAPSACK problem, there exists a PTAS constructed by Chekuri and Khanna (2006). Abrams et al. (2004) also provided an $(e/(e - 1))$ -approximation algorithm for their variant of SET k -COVER. For the winner determination problem in combinatorial auctions with submodular valuations, Feige and Vondrák (2006) developed an $(e/(e - 1) - \epsilon)$ -approximation algorithm

using a *demand oracle*, and Dobzinski and Schapira (2006) presented a $(2 - 1/n)$ -approximation algorithm using a *value oracle*. A demand oracle, given a set of prices p_i , one for each element e_i , outputs a set S that maximizes $f(S) - \sum_{e_i \in S} p_i$. This assumption of the existence of a demand oracle is a stronger assumption than that of a value oracle, as a demand oracle can simulate a value oracle in polynomial time (Dobzinski and Schapira (2006)). Recall that the existence of a value oracle is sufficient to show the factor-2 performance of a locally greedy algorithm for this problem. Nevertheless, these results do hint that there might be better approximation algorithms for the general class of problems, $f_S|_{\mathcal{F}_P}$, itself.

Calinescu et al. (2007) have recently developed an $(e/(e - 1))$ -approximation algorithm for the problem of maximizing the sum of weighted rank functions of matroids over an arbitrary matroid constraint. The sum of weighted rank functions are a rich subclass of monotone submodular functions, and include most of the objective functions of problem instances discussed in this work, but there do exist instances of monotone submodular functions that do not belong in the above class, notably including the objective function illustrated for the ADWORDS ASSIGNMENT problem in Lemma 2.

5. Generalized Results over Matroids and Independence Systems

An element e is said to be “admissible” into an independent set S if $S \cup \{e\}$ remains independent. Observe that for uniform matroids and partition matroids, the admissibility of an element into an independent set S depends only on the number of elements of each type present in S . Since the admissibility of an element into the greedy solution can be determined trivially for partition matroids and does not involve the need of an admissibility oracle, as would be the case for general matroids and independence systems, the above study was simple, and the running time of the greedy algorithms was independent of the size of the ground set, E . If a polynomial-time admissibility oracle does exist for a particular class of matroids or independence systems, then it is possible to study the performance of a greedy algorithm with an α -approximate incremental oracle for such a class of matroids.

Suppose that an independence system (E, \mathcal{F}) is the intersection of M different matroids. In this section, we shall generalize the result of Fisher et al. (1978), who proved that if an independence system (E, \mathcal{F}) is an intersection of a finite number, M , of matroids, then the standard greedy algorithm is a $(M + 1)$ -approximation algorithm. More formally, for the problem of maximizing a nondecreasing submodular function over (E, \mathcal{F}) , we shall show that a greedy algorithm, with an α -approximate incremental oracle as well as an admissibility oracle for (E, \mathcal{F}) at its disposal, is in fact an $(\alpha M + 1)$ -approximation algorithm.

We begin with a description of a generic greedy algorithm for this problem.

GREEDY ALGORITHM FOR $f_S|\mathcal{F}$

Initialization: Set $i := 1$; let $S_0 := \emptyset$, $E_0 := E$.

Step 1: If $E_{i-1} = \emptyset$, STOP.

Step 2: Select an element $e_i \in E_{i-1}$ for which $\alpha \rho_{e_i}(S_{i-1}) \geq \max_{e \in E_{i-1}} \rho_e(S_{i-1})$ using an α -approximate incremental oracle.

Step 3: Using the admissibility oracle, check if $S_{i-1} \cup \{e_i\} \in \mathcal{F}$.

Step 4a: If “no,” set $E_{i-1} := E_{i-1} \setminus \{e_i\}$ and return to **Step 1**.

Step 4b: Set $S_i := S_{i-1} \cup \{e_i\}$, $\rho_{i-1} := \rho_{e_i}(S_{i-1})$ and $E_i := E_{i-1} \setminus \{e_i\}$.

Step 5: Set $i := i + 1$ and return to **Step 1**.

Similar to the standard greedy algorithm described earlier for uniform matroids, the GREEDY ALGORITHM FOR $f_S|\mathcal{F}$ uses the α -approximate incremental oracle to select a candidate element of “good” incremental value. The algorithm then uses the admissibility oracle to check if the selected element is indeed admissible into the solution at hand, and modifies the ground set and solution set accordingly. Observe from the definition of an independence system that if an element e is not admissible to the candidate solution in an iteration i of the algorithm, then it is never admissible to the candidate solution after iteration i . Therefore, such elements may be removed from the underlying set for future consideration.

THEOREM 3. *Suppose (E, \mathcal{F}) is an independence system that can be expressed as the intersection of a finite number, M , of matroids, and f is a normalized, nondecreasing, submodular function. If z_g is the value of the greedy heuristic solution, utilizing an α -approximate incremental oracle and an admissibility oracle, for the following problem:*

$$\max\{f(S) : S \in \mathcal{F}\}$$

and z_{opt} is the value of an optimal solution, then $\frac{z_{opt}}{z_g} \leq \alpha M + 1$.

Proof. The proof presented here is an augmented version of the original proof by Fisher et al. (1978) for the case $\alpha = 1$, so as to overcome the difficulty that the sequence of incremental values ρ_i is, in general, not monotone anymore. Let us define U_t to be the set of elements considered in the first $(t + 1)$ iterations of the greedy algorithm before the addition of the $(t + 1)$ st element. Let $r_m(S)$ denote the rank of set S in matroid m (where the rank of S is the cardinality of the largest independent subset of S in the matroid), and $sp_m(S)$ be the span of S in matroid m , defined by:

$$sp_m(S) = \{e \in E : r_m(S \cup \{e\}) = r_m(S)\}.$$

In order to proceed with the proof, we shall utilize two lemmata shown by Fisher et al. (1978). We include the short proofs for the sake of completeness.

LEMMA 3 (Fisher et al. (1978)). $U_t \subseteq \bigcup_{m=1}^M sp_m(S_t) \quad t = 0, 1, \dots$

Indeed, if $j \in U_t$, then either $j \in S_t \subseteq \bigcup_{m=1}^M sp_m(S_t)$ for all m , or j is not admissible, implying that $j \in sp_m(S_t)$ for some matroid m .

LEMMA 4 (Fisher et al. (1978)). *If $\sum_{i=0}^{t-1} x_i \leq t$ for $t = 1, 2, \dots, K$, and $\rho_{i-1} \geq \rho_i$ with $\rho_i, x_i \geq 0$ for $i = 1, \dots, K - 1$ and $\rho_K = 0$, then $\sum_{i=0}^{K-1} \rho_i x_i \leq \sum_{i=0}^{K-1} \rho_i$.*

Consider the following linear program:

$$\max_x \left\{ \sum_{i=0}^{K-1} \rho_i x_i \mid \sum_{i=0}^{t-1} x_i \leq t, t = 1, \dots, K, \quad x_i \geq 0, i = 0, \dots, K-1 \right\}.$$

It is easy to verify that its dual is:

$$\min_z \left\{ \sum_{t=1}^K t z_{t-1} \mid \sum_{t=i}^{K-1} z_t \geq \rho_i, i = 0, \dots, K-1, \quad z_t \geq 0, t = 0, \dots, K-1 \right\}.$$

As $\rho_i \geq \rho_{i+1}$, the solution $z_i = \rho_i - \rho_{i+1}, i = 0, \dots, K-1$ (where $\rho_K = 0$) is dual feasible with value $\sum_{t=1}^K t(\rho_{t-1} - \rho_t) = \sum_{t=0}^{K-1} \rho_t$. By weak LP duality, the claim follows.

So, suppose that S and T represent the greedy and an optimal solution, respectively, to the above problem. Additionally, let $|S| = K$. Note that since (E, \mathcal{F}) is an independence system and not necessarily a matroid, $|T|$ need not be K .

For $t = 1, \dots, K$, let $s_{t-1} = |T \cap (U_t \setminus U_{t-1})|$, where U_t is the set of elements considered in the first $(t+1)$ iterations before the addition of a $(t+1)$ st element to S_t . We assume without loss of generality that $U_0 = \emptyset$ and $U_K = E$. Also, let $\rho^*(S_i) = \max_{e \in E_i} \rho_e(S_i)$ for $i = 0, \dots, K$.

Since f is a nondecreasing submodular set function, Lemma 1 yields:

$$z_{opt} = f(T) \leq f(S) + \sum_{e \in T \setminus S} \rho_e(S). \quad (6)$$

Suppose $t \in \{1, 2, \dots, K\}$ and $\rho_{q(t)} = \min\{\rho_i \mid i = 0, \dots, t-1\}$. Now for all elements $e \in T \cap (U_t \setminus U_{t-1})$, we have that:

$$\rho_e(S) \leq \rho_e(S_t) \leq \rho^*(S_t) \leq \rho^*(S_{q(t)+1}) \leq \alpha \rho_{q(t)}. \quad (7)$$

While the first inequality follows from the submodularity of f , the second and third follow from the definition of ρ^* . The final inequality follows from the fact that we are using an α -approximate oracle: If e^* is such that $\rho_{e^*}(S_{q(t)+1}) = \rho^*(S_{q(t)+1})$, then by the fact that $e^* \in E_{q(t)+1}$ was not considered by the greedy algorithm in iteration $q(t)$, the inequality follows. Given the above inequality, define $\rho'_{t-1} = \alpha \rho_{q(t)}$. We then have:

$$\rho_e(S) \leq \rho'_{t-1} \quad \text{for all } e \in T \cap (U_t \setminus U_{t-1}), \quad t = 1, 2, \dots, K. \quad (8)$$

Note that by the way that we have defined ρ'_{t-1} , it has the nonincreasing property. In other words, $\rho'_{t-1} \geq \rho'_t$ for all t . This is based on the definition of $q(t)$, which itself has the same nonincreasing property. Additionally, $\rho'_t = \alpha \rho_{q(t+1)} \leq \alpha \rho_t$.

Using this fact in equation (6), we now have that:

$$\begin{aligned} f(T) &\leq f(S) + \sum_{e \in T \setminus S} \rho_e(S) \\ &\leq f(S) + \sum_{e \in T} \rho_e(S) \\ &= f(S) + \sum_{t=1}^K \sum_{e \in T \cap (U_t \setminus U_{t-1})} \rho_e(S) \\ &\leq f(S) + \sum_{t=1}^K \rho'_{t-1} s_{t-1} \end{aligned} \quad (9)$$

where the last inequality follows from (8) and the definition of s_t .

Now, observe that since $s_{t-1} = |T \cap (U_t \setminus U_{t-1})|$, it must be that $\sum_{i=1}^t s_{i-1} = |T \cap U_t|$. By Lemma 3, we also have that $U_t \subseteq \bigcup_{m=1}^M sp_m(S_t)$, which in turn gives us that:

$$|T \cap U_t| \leq \sum_{m=1}^M |T \cap sp_m(S_t)| .$$

But since T is independent in each of the matroids and $r_m(sp_m(S_t)) = t$, it follows that for each m , $|T \cap sp_m(S_t)| \leq t$. This implies that:

$$\sum_{i=1}^t s_{i-1} \leq \sum_{m=1}^M |T \cap sp_m(S_t)| \leq Mt \quad (10)$$

where the above inequality is true for each $t = 1, 2, \dots, K$.

Since $\rho'_t, s_t \geq 0$ for all t , and ρ'_t has the nonincreasing property, by substituting $x_i = s_i/M$ and $\rho_i = \rho'_i$ in Lemma 4, inequality (10) now gives us that

$$\sum_{i=0}^{K-1} \rho'_i s_i \leq M \sum_{i=0}^{K-1} \rho'_i . \quad (11)$$

Substituting back inequality (11) into inequality (9), we now have that:

$$\begin{aligned} f(T) &\leq f(S) + M \sum_{i=0}^{K-1} \rho'_i \\ &\leq f(S) + \alpha M \sum_{i=0}^{K-1} \rho_i \\ &= f(S)(1 + \alpha M) \end{aligned} \quad (12)$$

where inequality (12) follows from the fact that $\rho'_t \leq \alpha \rho_t$ for all t . \square

In parallel to this work, Calinescu et al. (2007) have also recently noted a somewhat different proof for the performance of the greedy algorithm for $f_S|_{\mathcal{F}}$ in the presence of an α -approximate incremental oracle. In the next section, we offer further insights into the greedy algorithm and its performance.

5.1. Discussion on the Running Time of the Greedy Algorithm

As we have noted in earlier sections for uniform and partition matroids, the greedy algorithm and the locally greedy algorithm have a running time that depends only on the restrictions of the number of elements that must be picked, and is independent of the size of the ground set, E . However, the greedy algorithm presented for $f_S|_{\mathcal{F}}$ in fact has a running time that depends on the size of E . Hence, if E is exponentially large, this would yield a poor running time for the greedy algorithm. It must also be noted that this dependence of the running time on $|E|$ only comes because sometimes, the incremental oracle might pick an element $e \in E$ that need not be admissible. If however, there were a hybrid incremental oracle that always finds a “good” incremental element that is necessarily admissible, then the running time performance of the greedy algorithm will not necessarily depend on $|E|$, but on the size of a largest independent set in E . In certain instances of $f_S|_{\mathcal{F}}$, such as when the independence system (E, \mathcal{F}) is a matroid, it may be possible to bound the size of the largest independent set polynomially in the size of the input of the underlying combinatorial optimization problem. We now present an instance of such a problem, and as it turns out, the greedy algorithm consequently provides the best-known approximation ratio for this problem in polynomial time.

5.2. k -Median with Hard Capacities and Packing Constraints

Fleischer et al. (2006) presented the following variant of the k -Median problem, that they call the k -MEDIAN WITH HARD CAPACITIES AND PACKING CONSTRAINTS problem.

k -MEDIAN WITH HARD CAPACITIES AND PACKING CONSTRAINTS

Instance: A set, U , of n items and a set, B , of m bins. Each bin $i \in B$ has an independence system \mathcal{I}_i of subsets of items that fit in bin i . A profit p_{ij} for assigning item j to bin i . An integer $k \leq m$.

Task: Choose a set of K bins, $|K| \leq k$, and a subset of items, $S \subseteq U$, with a feasible assignment of these items to the bins in K , $S_i \in \mathcal{I}_i$ for bin $i \in K$, $S_i \cap S_l = \emptyset$ for $i \neq l$, so as to maximize profit, $\sum_{i \in K} \sum_{j \in S_i} p_{ij}$.

This problem has very similar flavor to the SEPARABLE ASSIGNMENT problem discussed in Section 2. In fact, using the transformation of Observation 1, it can be seen that the underlying constraint structure is a laminar matroid, defined by $E = \cup_{i=1}^m E_i$, and

$$\mathcal{F} = \{F : F \subseteq E, |F \cap E_i| \leq 1 \text{ for } i = 1, \dots, m \text{ and } |F| \leq k\}.$$

In the above matroid, it is easy to see that the size of the largest independent set (or basis of the matroid) is k , which is polynomial in the input (since $k \leq m$). Moreover, the objective function for this problem can be rewritten exactly as in Observation 1. Consequently, monotone submodularity follows. Thus k -MEDIAN WITH HARD CAPACITIES AND PACKING CONSTRAINTS is an instance of $f_S | \mathcal{F}_M$.

Fleischer et al. (2006) devised a polynomial-time local search $(\alpha + 1 + \epsilon)$ -approximation algorithm for k -MEDIAN WITH HARD CAPACITIES AND PACKING CONSTRAINTS, assuming there is an α -approximation algorithm for the single bin subproblem. The authors also remark that this result is, to the best of their knowledge, the first constant-factor approximation to this problem.

Given an α -approximation algorithm for each of the single bin subproblems corresponding to the m bins, one may easily design an α -approximate *hybrid* incremental oracle over all element types. At the start of any iteration i , suppose that the current solution generated by the greedy algorithm is S . If $|S| \leq k - 1$, then selecting the α -best incremental element among the l element types corresponding to the l bins for which a feasible packing has not been selected as yet, would indeed be a feasible selection. Hence the running time of a greedy algorithm for this problem is polynomial in the input size. Specifically, if P_i is the running time of the α -approximate oracle corresponding to bin i , then the running time of the algorithm is $O(k \sum_{i=1}^m P_i)$. Since the problem is an instance of $f_S | \mathcal{F}_M$ with a hybrid incremental oracle available, Theorem 3 implies:

COROLLARY 3. *Given an α -approximation algorithm for the single bin subproblem, there is a polynomial-time $(\alpha + 1)$ -approximation greedy algorithm for k -MEDIAN WITH HARD CAPACITIES AND PACKING CONSTRAINTS.*

Hence, by generalizing the results of Fisher et al. (1978), we are able to improve upon the previous best-known result of Fleischer et al. (2006) for this problem. In a related context, Calinescu et al. (2007) have developed an improved $e/(e - 1)$ -approximation algorithm for the GENERALIZED ASSIGNMENT problem subject to a laminar matroid constraint on the bins.

6. Concluding Remarks and Open Questions

In this paper, we extend some classic results of Fisher et al. (1978) and Nemhauser et al. (1978) on the performance of the greedy algorithm for maximizing monotone submodular functions over independence systems and other special subclasses. Our work is based on the premise that the greedy algorithm need not always to be able to pick an element of maximum incremental value, and may only be able to select an element of “good” incremental value. We show that this is indeed the case by posing some interesting and important discrete optimization problems as the problem of maximizing a monotone submodular function over an independence system. Based on our generalized results, we are able to reinterpret as well as present a new view to many recent results. In certain cases, we are even able to establish improved approximation results based on these insights.

We conclude by highlighting some interesting open questions that remain intimately connected to this work and even served to motivate this study of submodular function maximization:

- Consider the problem $f_S|\mathcal{F}_M$, of maximizing a submodular function over an arbitrary matroid constraint, given a value oracle. Calinescu et al. (2007) have recently conjectured that there exists an $e/(e-1)$ -approximation problem for this problem. For a rich subclass of submodular functions, Calinescu et al. (2007) show that there is indeed such an algorithm, based on pipage rounding and considering an appropriate extension of a submodular function. We believe that a insightful first step in proving the conjecture for $f_S|\mathcal{F}_M$ would be to develop an $e/(e-1)$ -approximation algorithm to $f_S|\mathcal{F}_P$, the problem of maximizing a submodular function over a partition matroid, given the special simple structure of the partition matroid.

- Consider the standard greedy and local greedy algorithms described for $f_S|\mathcal{F}_P$. One would note that in the tight worst-case examples described in Observation 2, the greedy algorithms’ bad performance may be attributed to the greedy algorithm not selecting the “correct” optimal incremental element in the first iteration. While simple randomizing over all optimal incremental elements in any iteration does not necessarily improve the performance of the greedy algorithm, smarter randomized schemes might lead to improved approximation algorithms for $f_S|\mathcal{F}_P$. Indeed, Goundan and Schulz (2007a) propose an improved randomized $(2 - 1/n)$ -approximation algorithm for $f_S|\mathcal{F}_P$. An interesting question is therefore if one might leverage an intermediate randomized scheme coupled with a greedy strategy to develop improved approximation results for $f_S|\mathcal{F}_P$.

Appendix A: Winner Determination in Combinatorial Auctions.

Combinatorial auctions are mechanisms via which multiple non-identical items are sold to bidders who express preferences over combinations of items, and not just single items. Such auctions assume particular relevance when the items being sold are either *complements* or *substitutes* to each other. In particular, Lehmann et al. (2006) studied the problem of an auctioneer who would like to allocate a set of items, X , of decreasing marginal values amongst n *submodular* bidders so as to maximize total social welfare. More formally, the problem may be stated as follows:

WINNER DETERMINATION

Instance: A set X of items; n bidders, each bidder j having a submodular valuation function, $v_j : 2^X \rightarrow \mathbb{R}_{\geq 0}$, which is normalized and nondecreasing.

Task: Find a partition of the items in X into pairwise disjoint sets, S_1, \dots, S_n , so as to maximize $\sum_{j=1}^n v_j(S_j)$.

It has been observed in the literature that the WINNER DETERMINATION problem is a special case of $f_S|\mathcal{F}_P$. We present this transformation here for completeness. Consider a ground set given by $E = \cup_{i \in X} E_i$, where an element $e_{ij} \in E_i$ corresponds to allocating item i to bidder j . The constraint defined by the WINNER

DETERMINATION problem, that any item $i \in X$ may be assigned to at most one bidder, would therefore transform to picking at most one element from each set E_i . Clearly, the set of all feasible subsets of E defined by this constraint would therefore be a partition matroid. Moreover, the objective function of the auctioneer is to maximize social utility, $\sum_{j=1}^n v_j(S_j)$, where (S_1, \dots, S_n) is a partition of X . This objective is also nondecreasing and submodular, since it is a sum of nondecreasing submodular functions. Based on any set $F \subseteq E$, define

$$S_j^F = \{i | e_{ij} \in F\}.$$

Furthermore, the objective function of the auctioneer may be rewritten as $f(F) = \sum_{j=1}^n v_j(S_j^F)$, which is clearly monotone submodular on the base set, E . This may be easily verified by noting that the marginal value of any element e_{ij} to a set F is indeed nonincreasing, since the marginal value of the corresponding item i being allocated to bidder j is itself nonincreasing. Thus the WINNER DETERMINATION problem is indeed an instance of $f_S | \mathcal{F}_P$, as noted by Lehmann et al. (2006).

Interestingly, we point out that the factor-2 greedy approximation algorithm proposed by Lehmann et al. (2006) for this problem turns out to be exactly the locally greedy algorithm of Fisher et al. (1978), as both algorithms are independent of item ordering. To complete the analogy between both these greedy algorithms, we observe that Lehmann et al. (2006) assume access to a *value oracle* for each player, to encode the submodular valuations of the players. That is, given a set S of items, the value oracle for bidder j outputs $v_j(S)$. In the corresponding locally greedy algorithm, the existence of an optimal incremental oracle follows from the existence of the value oracle, and from the fact that there are only n bidders whose valuations need to be checked to find the best incremental element. The authors do not make this connection and instead claim that the family of greedy algorithms that they consider is wider than that of Fisher et al. (1978).

One might observe that in the transformation described earlier in this section, the structure of the submodular objective function is “separable,” in the sense that the marginal utility of any item e_{ij} depends only on which other elements have been allocated to bidder j , i.e., elements of the form e_{kj} in the current solution, $F \subseteq E$. It is conceivable that one might leverage this special structure to devise improved approximation algorithms for the WINNER DETERMINATION problem. Indeed, Dobzinski and Schapira (2006) proposed a randomized $(2 - 1/n)$ -approximation algorithm for this problem, where n represents the number of elements of each type. Goundan and Schulz (2007a) have recently shown that this algorithm may be adapted to provide a $(2 - 1/n)$ -approximation algorithm for $f_S | \mathcal{F}_P$ as well. Interestingly, Khot et al. (2005) proved that there can be no polynomial time approximation algorithm with a factor better than $e/(e - 1)$ for the WINNER DETERMINATION problem, unless $P=NP$.

Appendix B: Convergence Issues in Competitive Games.

Vetta (2002) studied the following strategic game played amongst n players: associated with each player j is a disjoint ground set V_j of *actions*, and S_j , a collection of subsets of V_j . Any set $s_j \in S_j$ corresponds to a feasible *strategy* of player j . In addition, suppose that $\emptyset_j \in S_j$ corresponds to the null strategy for player j . A strategy profile or *state*, $S = (s_1, \dots, s_n)$, represents the corresponding strategies being played by each player. Let $S \oplus s'_j = (s_1, \dots, s_{j-1}, s'_j, s_{j+1}, \dots, s_n)$ denote the state obtained if player j were to change its strategy to s'_j . Assume that $\alpha_j : \Pi_j S_j \rightarrow \mathbb{R}$ represents the private utility function of player j , and $\gamma : \Pi_j S_j \rightarrow \mathbb{R}$, the social objective function. Suppose that the social objective function, $\gamma(\cdot)$, is a monotone submodular set function defined on $\cup_{j=1}^n V_j$, i.e.,

$$\gamma(S) = g(\cup_{j=1}^n s_j),$$

where g is a monotone submodular function defined on $\cup_j V_j$. Based on different assumptions on γ and α_j , Vetta (2002) introduced the following types of games:

Utility Game: A strategic game as described above is said to be a utility game if it satisfies the *Vickrey condition*:

$$\alpha_j(S) \geq \gamma(S) - \gamma(S \oplus \emptyset_j) \text{ for all feasible states, } S.$$

Valid Utility Game: A valid utility game is a utility game that satisfies the *Cake condition*:

$$\sum_j \alpha_j(S) \leq \gamma(S) \text{ for all feasible states, } S.$$

Basic Utility Game: A basic utility game is a valid utility game that satisfies the Vickrey condition with equality:

$$\alpha_j(S) = \gamma(S) - \gamma(S \oplus \emptyset_j) = g(\cup_i s_i) - g(\cup_{i \neq j} s_i).$$

We reinterpret valid utility games, and in particular, basic utility games, as an equivalent *decentralized approach* to maximizing a submodular function over a partition matroid, where at most one element may be picked of each type. We illustrate this equivalence for basic utility games, and the equivalence for valid utility games follows similarly.

Corresponding to an instance of a basic utility game, construct a ground set $E = \cup_{j=1}^n E_j$ and add an element e_j in E_j corresponding to each feasible strategy $s_j \in S_j$ of player j . That a player may select at most one strategy from its feasible set of strategies would lead to a natural partition matroid on this underlying ground set, E . Correspondingly, a function, f , may be defined on any subset $F \subseteq E$ as follows:

$$f(F) = g(\cup_{e_j \in F} s_j).$$

It may be verified that f is monotone submodular on E , since g itself is monotone submodular on the underlying ground set $\cup_j V_j$. Observe that in the transformation so far, we have made no assumptions whatsoever regarding the underlying private utilities of the players. Thus, by imbuing each player with any private utility function, we may define a corresponding *partition matroid game*, where E_j would correspond to the strategy space of player j . Indeed, if the private utility of each player in the partition matroid game is set to

$$\alpha_j(F) = f(F) - f(F \setminus E_j) \text{ for any } F \subseteq E,$$

then this partition matroid game defined would in fact be the basic utility game we sought to transform, since this private utility matches the private utility of player j in the basic utility game.

Conversely, starting with any instance of $f_S | \mathcal{F}_P$ wherein at most one element may be picked from each type, we may similarly define a partition matroid game over the instance. By imbuing the player representing elements of type j with the private utility function,

$$\alpha_j(F) = f(F) - f(F \setminus E_j) \text{ for any } F \subseteq E,$$

we clearly satisfy the Vickrey condition with equality. Moreover, it is not hard to verify that these utilities also satisfy the Cake condition (refer Theorem 2.5 of Vetta (2002)). Thus, we may define a basic utility game corresponding to each instance of $f_S | \mathcal{F}_P$ as well. Additionally, by defining alternate private utilities in the partition matroid game, one may draw a similar correspondence to valid utility games. Via this correspondence, we may now reinterpret the results of Vetta (2002) for valid utility games as the performance bounds of a decentralized approach to $f_S | \mathcal{F}_P$.

One of the main results of Vetta (2002) (Theorem 3.4) is that there exists a pure Nash equilibrium in any valid utility game, and that the expected social value of any (pure or mixed strategy) Nash equilibrium is at least half the social optimal value. This result may alternately be interpreted as:

COROLLARY 4. *Any Nash equilibrium of a decentralized valid-utility game approach to $f_S | \mathcal{F}_P$ is a factor 2 approximation to the optimal solution.*

Vetta (2002) gave examples that imply that this factor 2 result is indeed tight. Unfortunately, Goemans et al. (2005) show that for some instances of valid utility games (alternately, in a valid-utility game approach for certain instances of $f_S | \mathcal{F}_P$), finding a Nash equilibrium is PLS-complete.

Interestingly, iterative improved response strategies in a valid-utility game framework for $f_S | \mathcal{F}_P$ closely resemble local search approaches to $f_S | \mathcal{F}_P$. Indeed, Fisher et al. (1978) gave similar bounds on the performance of an *interchange heuristic*, a local improvement procedure. In an iteration of the interchange heuristic, while there exists an element e outside the current solution, S , that may be swapped with an element in S so as to improve the value of the solution while maintaining independence simultaneously, S is modified by interchanging the elements accordingly. The heuristic terminates when no feasible improving element remains in the “swap” neighborhood. Fisher et al. (1978) showed that a locally optimal solution obtained using the interchange heuristic is a 2-approximate solution to $f_S | \mathcal{F}_P$ (and more generally, over arbitrary matroids). Any locally optimal solution in a “swap” neighborhood to $f_S | \mathcal{F}_P$ in fact corresponds to a pure-strategy Nash equilibrium in a basic-utility game. The result of Fisher et al. (1978) implies that any pure-strategy Nash equilibrium in a basic utility game is at least half of the social optimal value. The

result of Vetta implied in Corollary 4 is more general, in that it holds for valid utility games and for mixed strategy Nash equilibria as well, although the structure of both proofs are similar in spirit.

Mirroknii and Vetta (2004) also considered the notion of a state graph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ corresponding to a utility game, where each vertex in \mathcal{V} represents a strategy state, $S = (s_1, \dots, s_n)$. There is a directed edge in \mathcal{E} from state S to S' with label j if the only difference between S and S' is the strategy of player j ; and player j plays its best response in strategy state S to go to S' . A one-round best-response path is a path P that starts from an arbitrary state and the edges of P are labeled in order i_1, i_2, \dots, i_n , where i_1, i_2, \dots, i_n is an arbitrary ordering of the n players. One may easily define a similar state graph and related notions for a partition matroid game.

We claim that starting with an initial state $(\emptyset_1, \emptyset_2, \dots, \emptyset_n)$ in the state graph of a basic utility game and following a one-round best response path corresponds to the execution of the locally greedy algorithm of the underlying partition matroid. To see this, without loss of generality, one may assume that the best response path in consideration is labeled $1, 2, \dots, n$. Furthermore, let the vertices in this path correspond to $S_0 = (\emptyset_1, \emptyset_2, \dots, \emptyset_n)$, $S_1 = (s_1, \emptyset_2, \dots, \emptyset_n)$, \dots , $S_n = (s_1, s_2, \dots, s_n)$ in order. Now, in any iteration j of the locally greedy algorithm, the role of the incremental oracle is to pick an element $e \in E_j$ of maximum possible incremental value, $\rho_e(F_{j-1}) = f(F_{j-1} \cup e) - f(F_{j-1})$, to the current solution at hand, F_{j-1} . In the one-round best response path being considered, the social objective at vertex S_{j-1} is given by $g(\cup_{i=1}^{j-1} s_i)$. By induction, suppose that $F_{j-1} = \{e_i | i = 1, \dots, j-1\}$ where each e_i corresponds to the strategy s_i of player i in state $S_{j-1} = (s_1, \dots, s_{j-1}, \emptyset_j, \dots, \emptyset_n)$. Clearly by definition,

$$f(F_{j-1}) = g(\cup_{i=1}^{j-1} s_i) = \gamma(S_{j-1}).$$

Moreover, in transitioning from S_{j-1} to S_j , player j selects s so as to maximize

$$\alpha_j(T) = g(\cup_{i=1}^{j-1} s_i \cup s) - g(\cup_{i=1}^{j-1} s_i),$$

where $T = (s_1, \dots, s_{j-1}, s, \dots, \emptyset)$. However, observe that

$$\alpha_j(T) = f(F_{j-1} \cup e) - f(F_{j-1}) = \rho_e(F_{j-1}),$$

where e would be the element in E_j corresponding to the strategy s . Hence, it must be that the element selected by the locally greedy algorithm is indeed the e_j that corresponds to s_j . The claim follows by induction.

Mirroknii and Vetta (2004) showed that a one-round best response path starting from the initial state, $(\emptyset_1, \emptyset_2, \dots, \emptyset_n)$, provides a 2-approximation to the state S that maximizes $\gamma(S)$. By our interpretation of this path as the execution of the locally greedy algorithm, the same result follows from Theorem 2.

Appendix C: Set k -Cover Problems in Wireless Sensor Networks

Motivated by applications in wireless sensor networks, Abrams et al. (2004) considered the following variant of the SET k -COVER problem:

SET k -COVER

Instance: A set of elements, U , a collection \mathcal{S} of subsets of U , and an integer $k \geq 2$.

Task: Find a partition of the collection of subsets \mathcal{S} into k parts, $\mathcal{C}_1, \dots, \mathcal{C}_k$, such that $\sum_{i=1}^k |\cup_{S_j \in \mathcal{C}_i} S_j|$ is maximized.

The intuition behind the formulation of this problem is as follows: the underlying elements of the set U are meant to represent distinct regions being monitored by a sensor network, and each subset $S_i \in \mathcal{S}$ represents the regions monitored by a particular wireless sensor i . The objective of the planner is to partition these sensors into k parts so as to maximize the number of times the regions are covered by these parts. Each part of the partition corresponds to a group of sensors that are activated for a particular period of time, and different parts of the partition are activated at different times, so as to conserve the battery power of the wireless sensors.

Consider a ground set $E = \cup_{j=1}^{|S|} E_j$, where any element $e_{ij} \in E_j$ corresponds to assigning set S_j to partition C_i . That a set in S may be allocated to at most one partition defines a partition matroid on E . Moreover, for any subset $F \subseteq E$, create a partition with $C_i^F = \{S_j | e_{ij} \in F\}$. Now, the objective function of the SET k -COVER problem corresponds to:

$$f(F) = \sum_{i=1}^k |\cup_{S_j \in C_i^F} S_j|.$$

It is not hard to see that f is nondecreasing and submodular, using a similar argument as seen for MAXIMUM COVERAGE. Therefore, this problem is an instance of $f_S | \mathcal{F}_P$. Abrams et al. (2004) proposed a number of algorithms, including a distributed greedy algorithm, and showed that it is a 2-approximation algorithm for the problem. This distributed greedy algorithm is in fact analogous to the locally greedy algorithm, and the performance of the distributed greedy algorithm of Abrams et al. (2004) follows from Theorem 2.

References

- Abrams, Z., A. Goel, S. Plotkin. 2004. Set k -cover algorithms for energy efficient monitoring in wireless sensor networks. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. 424–432.
- Calinescu, G., C. Chekuri, M. Pál, J. Vondrák. 2007. Maximizing a submodular set function subject to a matroid constraint. *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, vol. 4513. 182–196.
- Chekuri, C. 2006. Cited as personal communication in Fleischer et al. (2006).
- Chekuri, C., S. Khanna. 2006. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing* **35** 713–728.
- Chekuri, C., A. Kumar. 2004. Maximum coverage problem with group budget constraints and applications. *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Lectures Notes in Computer Science*, vol. 3122. 72–83.
- Conforti, M., G. Cornuéjols. 1984. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics* **7** 251–274.
- Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, A. Schrijver. 1998. *Combinatorial Optimization*. Wiley-Interscience.
- Dawande, M., J. Kalagnanam, P. Keskinocak, R. Ravi, F. S. Salman. 2000. Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization* **4** 171–186.
- Dobzinski, S., M. Schapira. 2006. An improved approximation algorithm for combinatorial auctions with submodular bidders. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1064–1073.
- Edmonds, J. 1971. Matroids and the greedy algorithm. *Mathematical Programming* **1** 127–136.
- Farahat, A., C. Barnhart. 2004. Improved bounds and a generalized greedy algorithm for maximizing submodular functions over matroids. Working Paper, Massachusetts Institute of Technology.
- Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM* **45** 634–652.
- Feige, U., J. Vondrák. 2006. Approximation algorithms for allocation problems: Improving the factor of $1 - 1/e$. *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*. 667–676.
- Fisher, M. L., G. L. Nemhauser, L. A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming Study* **8** 73–87.
- Fleischer, L. K., M. X. Goemans, V. S. Mirrokni, M. Sviridenko. 2006. Tight approximation algorithms for maximum general assignment problems. *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*. 611–620.
- Fujishige, S. 2005. *Submodular Functions and Optimization*. Elsevier.

- Gabow, H., T. Kohno. 2001. A network-flow-based scheduler: Design, performance history, and experimental analysis. *Journal of Experimental Algorithmics* **6** 1–40.
- Goemans, M. X., V. S. Mirrokni, A. Vetta. 2005. Sink equilibria and convergence. *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. 142–154.
- Goundan, P. R., A. S. Schulz. 2007a. An improved approximation algorithm for maximizing a submodular set function over a partition matroid. Working Paper, Massachusetts Institute of Technology.
- Goundan, P. R., A. S. Schulz. 2007b. A note on scheduling advertisements on a web page. Working Paper, Massachusetts Institute of Technology.
- Hochbaum, D. S., ed. 1997. *Approximation Algorithms for NP-Hard Problems*. PWS.
- Hochbaum, D. S., A. Pathria. 1998. Analysis of the greedy approach in problems of maximum k -coverage. *Naval Research Logistics* **45** 615–627.
- Jansen, K., G. Zhang. 2004. On rectangle packing: Maximizing benefits. *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*. 204–213.
- Khot, S., R. J. Lipton, E. Markakis, A. Mehta. 2005. Inapproximability results for combinatorial auctions with submodular utility functions. *Proceedings of the 1st International Workshop on Internet and Network Economics*. 92–101.
- Khuller, S., A. Moss, J. S. Naor. 1999. The budgeted maximum coverage problem. *Information Processing Letters* **70** 39–45.
- Korte, B., D. Hausmann. 1978. An analysis of the greedy algorithm for independence systems. B. Alspach, P. Hell, D. J. Miller, eds., *Algorithmic Aspects of Combinatorics, Annals of Discrete Mathematics*, vol. 2. North-Holland, 65–74.
- Korte, B., J. Vygen. 2000. *Combinatorial Optimization: Theory and Algorithms*. Springer.
- Lawler, E. L. 1976. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston.
- Lehmann, B., D. Lehmann, N. Nisan. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* **55** 270–296.
- Lovász, L. 1983. Submodular functions and convexity. A. Bachem, M. Grötschel, B. Korte, eds., *Mathematical Programming: The State of the Art*. Springer, 234–257.
- Mirrokn, V. S., A. Vetta. 2004. Convergence issues in competitive games. *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Lecture Notes in Computer Science*, vol. 3122. 183–194.
- Nemhauser, G. L., L. A. Wolsey. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research* **3** 177–188.
- Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley-Interscience.
- Nemhauser, G. L., L. A. Wolsey, M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming* **14** 265–294.
- Rado, R. 1957. Note on independence relations. *Proceedings of the London Mathematical Society*, vol. 7. 300–320.
- Schrijver, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.
- Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* **32** 41–43.
- Vetta, A. 2002. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*. 416–425.
- Vohra, R. V., N. G. Hall. 1993. A probabilistic analysis of the maximal covering location problem. *Discrete Applied Mathematics* **43** 175–183.
- Wolsey, L. A. 1982. Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Mathematics of Operations Research* **7** 410–425.