DISSERTATION

PERFORMANCE BOUNDS FOR GREEDY STRATEGIES IN SUBMODULAR OPTIMIZATION PROBLEMS

Submitted by

Yajing Liu

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2018

Doctoral Committee:

Advisor: Edwin K. P. Chong
Co-Advisor: Ali Pezeshki

J. Rockey Luo
Dan Bates

ABSTRACT

PERFORMANCE BOUNDS FOR GREEDY STRATEGIES IN SUBMODULAR OPTIMIZATION PROBLEMS

The greedy strategy is an approximate optimization algorithm which makes a locally optimal decision at each step. In many problems, the greedy strategy does not yield a globally optimal solution. How good is the greedy solution compared to the optimal solution? When the problem over matroid constraints has a property called submodularity, the greedy strategy is proved to produce a solution with value at least a constant scalar times the optimum value. In this thesis, we mainly investigate the performance of the greedy strategy in two classes of submodular optimization problems over matroid constraints. The first is set submodular optimization, which is to choose a set of actions to optimize a submodular objective function, and the second is string submodular optimization, which is to choose an ordered set of actions to optimize a string submodular objective function.

For set submodular optimization problems, we first provide performance bounds in terms of the total curvature for the batched greedy strategy under matroid constraints, where the greedy strategy is a special case with batch size equal to 1. Then we provide improved bounds for the greedy strategy by defining a partial curvature. Moreover, we use similar techniques for bounding the batched greedy strategy to provide performance bounds for social-aware Nash equilibria and group Nash equilibria in utility systems with user groups. For string submodular optimization problems, we first provide weakened sufficient conditions for the greedy strategy to be bounded by a scalar factor. Then based on the theory of string submodular functions, we develop a framework to bound the performance of approximate dynamic programming (ADP) schemes in path-dependent action optimization (PDAO) problems, where every control decision is treated as the solution to an optimization problem with a path-dependent objective function.

Consider the problem of choosing a set of actions to optimize an objective function that is set submodular. The batched greedy strategy is an approximation algorithm, which starts with the empty set, then iteratively adds to the current solution set a batch of elements that results in the largest gain in the objective function. We first investigate performance of the batched greedy strategy over the matroid constraints. To be specific, we develop bounds on the performance of the batched greedy strategy relative to the optimal strategy in terms of a parameter called the total batched curvature. We show that when the objective function is a polymatroid set function, the batched greedy strategy satisfies a harmonic bound for a general matroid constraint and an exponential bound for a uniform matroid constraint, both in terms of the total batched curvature. We also study the behavior of the bounds as functions of the batch size. Specifically, we prove that the harmonic bound for a general matroid is nondecreasing in the batch size and the exponential bound for a uniform matroid is nondecreasing in the batch size under the condition that the batch size divides the rank of the uniform matroid. Finally, we illustrate our results by considering a task scheduling problem and an adaptive sensing problem.

The greedy strategy is a special case of the batched greedy strategy with batch size equal to 1. The greedy strategy is known to satisfy some performance bounds in terms of the total curvature. The total curvature depends on function values on sets outside the constraint matroid. If the function is defined only on the matroid, the problem still makes sense, but the existing bounds involving the total curvature do not apply, which is puzzling. This motivates an alternative formulation of such bounds. The first question we address is whether it is possible to extend a polymatroid function defined on a matroid to one on the entire power set. This was recently shown to be negative in general. Here, we provide necessary and sufficient conditions for the existence of an incremental extension of a polymatroid function defined on the uniform matroid of rank k to one with rank $k+1$, together with an algorithm for constructing the extension. Whenever a polymatroid function defined on a matroid can be extended to the entire power set, the bounds involving the total curvature of the extension apply. However, these bounds still depend on sets outside the constraint matroid. Motivated by this, we define a new notion of curvature called partial curvature, involving only sets in the matroid. We derive necessary and sufficient conditions for an extension to have a total curvature equal to the partial curvature. Moreover, we prove that the bounds in terms of the partial curvature are in general improved over the previous ones. We illustrate our results with two contrasting examples motivated by practical problems.

We use the similar techniques for bounding the batched greedy strategy to bound the performance of Nash equilibria when there exists “grouping" in utility systems. We consider variations of the utility system considered by Vetta [1], in which users are grouped together. Our aim is to establish how grouping and cooperation among users affect performance bounds. We consider two types of grouping. The first type is from [2], where each user belongs to a group of users having social ties with it. For this type of utility system, each user's strategy maximizes its social group utility function, giving rise to the notion of social-aware Nash equilibrium. We prove that this social utility system yields to the bounding results of Vetta for non-cooperative system, thus establishing provable performance guarantees for the social-aware Nash equilibria. For the second type of grouping we consider, the set of users is partitioned into disjoint groups, where the users within a group cooperate to maximize their group utility function, giving rise to the notion of group Nash equilibrium. In this case, each group can be viewed as a new user with vector-valued actions, and a 1/2 bound for the performance of group Nash equilibria follows from the result of Vetta. But we derive tighter bounds involving curvature by defining the group curvature. Finally, we present an example of a utility system for database assisted spectrum access to illustrate our results.

Consider the problem of choosing a string of actions to optimize an objective function that is string submodular. Streeter and Golovin [3] show that if the objective function is prefix and postfix monotone and string submodular, then the greedy strategy achieves at least a $(1-1/e)$-approximation of the optimal strategy. Zhang et al. [4] consider a weaker notion of the postfix monotoneity and provide sufficient conditions for the greedy strategy to achieve a factor of at least $1-1/e$. We introduce the notions of K-submodularity and K-GO-concavity, which together are sufficient for this bound to hold, where K is the optimization horizon length. By introducing a notion of curvature η, we prove an even tighter bound with the factor $(1-e^{-\eta})/\eta$. Finally, we illustrate the strength of our results by considering two example applications. We show that our results provide weaker conditions on parameter values in these applications than in [4].

Based on the theory of string submodularity, we develop a framework to bound the performance of approximate dynamic programming (ADP). We consider a broad family of control strategies called path-dependent action optimization (PDAO), where every control decision is treated as the solution to an optimization problem with a path-dependent objective function. How well such a scheme works depends on the chosen objective function to be optimized and, in general, it might be difficult to tell, without doing extensive simulation and testing, if a given PDAO design gives good performance or not. We develop a framework to bound the performance of PDAO schemes, based on the theory of submodular functions. We show that every PDAO scheme is a greedy scheme for some optimization problem, and if that optimization problem is equivalent to our problem of interest and is provably submodular, then we can say that our PDAO scheme is no worse than something like $(1 - 1/e)$ of optimal. We show how to apply our framework to stochastic optimal control problems to bound the performance of ADP schemes. Such schemes are based on approximating the expected value-to-go term in Bellman's principle by computationally tractable means. Our framework provides the first systematic approach to bounding the performance of general ADP methods in the stochastic setting.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep appreciation to my advisors, Prof. Edwin K. P. Chong and Prof. Ali Pezeshki for accepting me as a Ph. D student, for training me in doing research, writing papers, and doing presentations, for encouraging me to broaden my horizons by attending conferences, summer schools, workshops, and doing internships, and for all the financial and spiritual support during my graduate studies at Colorado State University. It has been a great honor and pleasure to study and conduct research under the supervision of Profs Chong and Pezeshki. They are not only my mentors in research, but also in life! I would also like to express my sincere thanks to my committee members, Prof. J. Rockey Luo and Prof. Dan Bates for their important suggestions and remarks in my research and for their time and efforts in serving as my committee.

I would especially like to thank Dr. Zhenliang Zhang, Dr. Yang Zhang, Dr. Wenbing Dang, Dr. Shankarachary Ragi, Dr. Ramin Zahedi, and Dr. Pooria Pakrooh for sharing their research experiences and encouraging me to carry on through these years. Thanks go to the former and current members in our group, in particular, Yugandhar Sarkale, Fateh El Sherif, Somayeh Hosseini, Mahsa Ghorbani, Apichart Vasutapituks, Tushar Ganguli, Chris Robbiano, and Pranav Damale for their friendship and assistance on my research work. Thanks also go to my great friends Wenjing Liu, Xuemin Wang, Minda Le, Huan Wang, Zheng Wang, Xinran Xu, Yanru Tang, and Savini Samarasinghe for their encouragement in both research and life.

I acknowledge the people who mean a lot to me, my parents, Xinsheng Liu and Li Zhang, for working hard to support me to go to college and giving me liberty to choose what I desired; my twin-brother, Yabin Liu, and my sister in law, Pan Zhai for their selfless love and care to my parents which creates me an opportunity to concentrate on my research during my Ph.D study.

I owe many thanks to a very special person, my husband, Dr. Haonan Chen for his continued and unfailing love, support, understanding during my pursuit of Ph.D degree that made the completion of thesis possible.

DEDICATION

I would like to dedicate this dissertation to my family.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

We are often faced with choosing a small set of actions from a ground set of actions to optimize an objective function in real applications. A specific example is the task assignment problem, one of the fundamental combinatorial optimization problems in the branch of optimization or operations research. This problem has a number of agents and a number of tasks. Each agent can be assigned to perform any task with a given probability to accomplish the task. The aim is to choose a given number of agents to maximize the probability of accomplishing the tasks.

When the number of agents is not that large, we can use brute force method to enumerate all possible solutions and find the optimal solution. However, when the number of agents is large, it is impractical to enumerate all the possible solutions. At this point, we have to resort to approximation methods and one of the most popular approximation methods is the greedy strategy, which starts with the empty set, and iteratively adds to the current solution set an element that results in the largest gain in the objective function while satisfying the constraints. The greedy strategy yields a local optimal solution that approximates a globally optimal solution in a reasonable amount of time. The downside is that there is often no theoretical guarantee for the greedy strategy. But when the problem has a special property called submodularity, the greedy strategy is proved to produce a solution with value at least a constant scalar times the optimum value over matroid constraints. Celebrated results by Nemhauser et al. [5, 6] prove that when the objective function f is a monotone submodular set function with $f(\emptyset) = 0$, the greedy strategy yields a $1/2$-approximation[1] for a general matroid and a $(1 - e^{-1})$-approximation for a uniform matroid.

Submodularity is a property of set functions, whose value has the property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases. It is also called diminishing return property in eco-

[1]The term β-approximation means that $f(G)/f(O) \geq \beta$, where G and O denote a greedy solution and an optimal solution, respectively.

nomics, and has connectivity with both convexity and concavity [7]. It appears in a wide variety of applications such as viral marketing [8], information gathering [9], image segmentation [10], document summarization [11], feature selection [12], active learning [13], and sensor placement [14]. Therefore, the performance of approximation algorithms such as greedy schemes in submodular optimization problems has gained more attention in recent years [15].

In this thesis, we are interested in the performance of the greedy type schemes in submodular maximization problems and the specific topics are: the performance of the batched greedy strategy and more applicable performance bounds for the greedy strategy in submodular set optimization problems, the performance of the social-aware Nash equilibria and the group Nash equilibria in submodular utility systems, the performance of the greedy strategy in string submodular optimization problems, and the performance of approximate dynamic programming schemes in stochastic submodular control problems. We will introduce these topics specifically in the following section.

1.1 Background and Motivation

1.1.1 Batched Greedy Strategy in Set Optimization

A variety of combinatorial optimization problems such as generalized assignment (see, e.g., [3, 16–19]), welfare maximization (see, e.g., [20–22]), maximum coverage (see, e.g., [23–25]), maximal covering location (see, e.g., [26–29]), and sensor placement (see, e.g., [9, 30–32]) can be formulated as a problem of maximizing a set function subject to a matroid constraint. More precisely, the objective function maps the power set of a ground set to real numbers, and the constraint is that any feasible set is from a non-empty collection of subsets of the ground set satisfying matroid constraints.

Finding the optimal solution to the problem above in general is NP-hard. The greedy strategy provides a computationally feasible approach, which starts with the empty set, and then iteratively adds to the current solution set one element that results in the largest gain in the objective function, while satisfying the matroid constraints. This scheme is a special case of the batched greedy strategy with batch size equal to 1. For general batch size (greater than 1), the batched greedy strategy starts with the empty set but iteratively adds, to the current solution set, a batch of elements with the largest gain in the objective function under the constraints.

The performance of the batched greedy strategy with batch size equal to 1 has been extensively investigated in [5,6,33–36]. The performance of the batched greedy strategy for general batch size, however, has received little attention, notable exceptions being Nemhauser et al. [6] and Hausmann et al. [33]. Although Nemhauser et al. [6] and Hausmann et al. [33] investigated the performance of the batched greedy strategy, they only considered uniform matroid constraints and independence system constraints, respectively. This prompts us to investigate the performance of the batched strategy more comprehensively.

1.1.2 More Applicable Bounds for Greedy Strategy in Set Optimization

Conforti and Cornuéjols [34] define the total curvature to characterize the submodular property of the objective function, and they prove that the greedy strategy in set maximization problems satisfies some performance bounds in terms of the total curvature under matroid constraints when the objective function is a polymatroid set function. However, the total curvature depends on the function values on sets outside the matroid. This gives rise to the following issue when applying the existing bounding results involving the total curvature: If we are given an objective function defined only on the matroid, then the problem still makes sense, but the total curvature is no longer well defined. This means that the existing results involving the total curvature do not apply. But this surely is puzzling: if the optimization problem is perfectly well defined, why should the bounds no longer apply? This motivates us to investigate more applicable bounds involving only sets in the matroid.

1.1.3 Nash Equilibria in Utility Systems

A variety of interesting practical problems can be posed as utility maximization problems: these include facility location [37], traffic routing and congestion management [38] , sensor selection [39], and network resource allocation [2]. In a utility maximization problem, a set of users make decisions according to their own set of feasible strategies, resulting in an overall social utility value, such as profit, coverage, achieved data rate, and quality of service. The goal is to maximize the social utility function. Often, the users do not cooperate in selecting their strategies.

In general, it is impractical to find the globally optimal sequence (finite, ordered collection) of strategies maximizing the social utility function. Typically, it is more useful to consider scenarios where individual users or groups of users separately maximize their own private objective functions, and then ask how this compares with the globally optimal case. The usual framework for studying such scenarios is game theory together with its celebrated notion of Nash equilibria. A Nash equilibrium is a sequence of strategies (deterministic or randomized) for which no user can improve its own private utility by changing its strategy unilaterally. The question of how the Nash solution compares with the globally optimal solution is one of the most challenging problems in game theory. For a general utility maximization problem, [1] develops lower bounds on the worst-case social utility value in non-cooperative games.

With the advent of social networks, there is increasing interest in understanding the role of cooperation and social ties in games [40]. Motivated by the idea of bounding the batched greedy strategy, we are interested in exploring bounds for Nash equilibria when there exists "grouping" among users.

1.1.4 Greedy Strategy in String Optimization

In a variety of problems in engineering and applied science such as sequential decision making ([41–43]), adaptive sensing ([9,44]), and adaptive control ([45,46]), we are faced with optimally choosing a string (ordered set) of actions over a finite horizon to maximize an objective function under some constraints. We call this class of optimization string optimization.

The solution to the string optimization problems can be characterized using backward dynamic programming via Bellman's principle ([47,48]). However, dynamic programming is hard to implement because that the computational complexity of this approach grows exponentially with the size of action set and the horizon length. Hence, we often turn to approximation techniques. One approximation technique is the greedy strategy, which is to find an action at each stage to maximize the step-wise gain in the objective function. The performance for the greedy strategy in string optimization problems has been extensively investigated by [3] and [4]. Streeter and Golovin [3] proves that the greedy strategy satisfies a constant performance bound when the problem satisfies some properties. Zhang *et al.* [4] consider weaker conditions and provide a stronger bound by introducing curvature. But all the sufficient conditions obtained so far involve strings of length greater than K, even though the optimization problem involves only strings up to length K. This motivates a weakening of these sufficient conditions to involve only strings of length at most K, but still preserving the bounds here.

1.1.5 Approximate Dynamic Programming Schemes in Stochastic Control

We consider a broad family of control strategies that we call path-dependent action optimization (PDAO). To use a PDAO scheme is to treat every control decision as the solution to an optimization problem with a path-dependent objective function. How well such a scheme works depends on the chosen objective function to be optimized. A key result in optimal control theory is that, under quite general conditions, there exists an optimal solution (policy) that is also a PDAO scheme. This result, not usually stated this way and more commonly known as Bellman's principle, makes PDAO schemes of interest in a wide range of computational-intelligence applications and is the basis for self-driving vehicles and AlphaGo, the master-beating Go playing machine. Bellman's principle tells us that the path-dependent objective function to be optimized at each decision epoch must capture both the immediate reward as well as the (expected) long-term net reward associated with each candidate action. This embodies a rigorous notion of delayed gratification, common to all nontrivial optimal dynamic decision-making policies.

The key to the performance of a PDAO scheme is the design of good objective functions. The future-rewards part of the objective function prescribed by Bellman's principle, unfortunately, often cannot be computed exactly. Therefore, approximation methods are needed. These include a variety of approaches, ranging from reinforcement learning with deep neural networks to model-based Monte Carlo sampling (for an overview in the context of adaptive sensing, refer to [49]).

The family of PDAO schemes of interest here is often called approximate dynamic programming (ADP). Such schemes are based on approximating the second term on the right-hand side of Bellman's optimality principle (the expected value-to-go) by computationally tractable means. Although a wide range of approximate dynamic programming (ADP) methods have been developed [47–49], a general systematic technique to provide performance guarantees for them has remained elusive. This motivates us to derive performance bounds for general ADP methods in the stochastic setting.

1.2 Our Contributions

In Chapter 2, first we define the total k-batch curvature c_k and prove that when the objective function f is a polymatroid set function, the k-batch greedy strategy achieves a $1/(1+c_k)$-approximation for a general matroid and a $(1-(1-\frac{c_k}{l+1}\frac{m}{k})(1-\frac{c_k}{l+1})^l)/c_k$-approximation for a uniform matroid, where $K = kl + m$ is the rank of the uniform matroid, l and m are non-negative integers, and $0 < m \leq k$. When $c_k = 1$, the bound for a uniform matroid becomes $(1-(1-\frac{m}{k(l+1)})(1-\frac{1}{l+1})^l)$, which is the bound in [6]. When $k = 1$, the bound for a general matroid becomes $1/(1+c)$, which is the bound in [34], and the bound for a uniform matroid becomes $(1-(1-c/K)^K)/c$, which is the bound in [34]. When $m = k$, the bound for a uniform matroid becomes $(1-(1-c_k/(l+1))^{l+1})/c_k$, which is the bound in [50]. Then we prove that c_k is nonincreasing in k when f is a polymatroid set function. This implies that the larger the k, the better the harmonic bound for a general matroid and when k divides K, the larger the k, the better the exponential bound for a uniform matroid. Finally, we present a task scheduling problem and an adaptive sensing problem to demonstrate our results.

In Chapter 3, we first provide necessary and sufficient conditions for the existence of an extension of a polymatroid function f defined on the matroid to a polymatroid function g defined on the whole power set. Then, it follows that for problems satisfying the necessary and sufficient conditions, the greedy strategy satisfies the bounds $1/(1+d)$ and $(1-(1-d/K)^K)/d$ for a general matroid and a uniform matroid, respectively, where $d = \inf_{g\in\Omega_f} c(g)$ and Ω_f is the set of all polymatroid functions g on 2^X that agree with f on $\mathcal{I}$, i.e., $g(A) = f(A)$ for any $A \in \mathcal{I}$. These bounds apply to problems where the objective function is defined only on the matroid and satisfies the necessary and sufficient conditions. When the objective function is defined on the entire power set, it is clear that $d \leq c(f)$, which implies that the bounds are improved.

Next, we define a curvature b involving only sets in the matroid, and we prove that $b(f) \leq c(f)$ when f is defined on the entire power set. We derive necessary and sufficient conditions for the existence of an extended polymatroid function g such that $c(g) = b(f)$. This gives rise to improved bounds $1/(1 + b(f))$ and $(1 - (1 - b(f)/K)^K)/b(f)$ for a general matroid and a uniform matroid, respectively. Moreover, these bounds are not influenced by sets outside the matroid.

Finally, we present two examples. We first provide a task scheduling problem to show that a polymatroid function f defined on the matroid can be extended to a polymatroid function g defined on the entire power set while satisfying the condition that $c(g) = b(f)$, which results in a stronger bound. Then, we provide an adaptive sensing problem to show that there does not exist any extended polymatroid function g such that $c(g) = b(f)$ holds. However, for our extended polymatroid function g, it turns out that $c(g)$ is very close to $b(f)$ and much smaller than $c(f)$, which also results in a stronger bound.

In Chapter 4, we first describe the framework of [2] and show that a social-aware utility system yields to the bounding results of Vetta for non-cooperative system, thus establishing provable performance guarantees for the social-aware Nash equilibrium. Next, we describe our second type of grouping involving l disjoint groups with in-group cooperation. In this case, each group can be viewed as a new user with vector-valued actions, and a 1/2 bound for the performance of group Nash equilibrium follows from the result of [1]. We then define the group curvature c_{k_i} associated with group i with k_i users, and we show that if the social utility function is nondecreasing and submodular, then any group Nash equilibrium achieves at least $1/(1+\max_{1\leq i\leq l} c_{k_i})$ of the optimal social utility, which is tighter than that for the case without grouping. Especially, if each user has the same action space, then we have that any group Nash equilibrium achieves at least $1/(1 + c_{k^*})$ of the optimal social utility, where k^* is the least number of users among all the groups. In Section 5, we present an example of a utility system for database assisted spectrum access, adopted from [2]. We show that the utility system for this example is valid and the social utility function is submodular, illustrating an application of our results.

In Chapter 5, we introduce the notions of K-submodularity and K-GO-concavity, which together are sufficient for the $(1-(1-1/K)^K)$ bound to hold. By introducing a notion of curvature $\eta \in (0,1]$, we prove an even tighter bound with the factor $(1-e^{-\eta})/\eta$. Finally, we illustrate the strength of our results by considering two example applications. We show that our results provide weaker conditions on parameter values in these applications than in previous results reported in [4].

In Chapter 6, we develop a framework to bound the performance of ADP schemes. Our bounding method is based on the theory of submodular optimization [4]. The basic result from string submodular optimization is that every greedy scheme achieves at least $(1-1/e)$ of the optimum value. We first prove that every PDAO scheme is a greedy scheme for some optimization problem. If that optimization problem is equivalent to our problem of interest and is provably submodular (in a certain sense to be made precise later), then we can say with certainty that our PDAO scheme is no worse than $(1-1/e)$ of optimal. We then show how to apply our framework to bound ADP schemes in stochastic optimal control problems Markov decision processes (MDPs). ADP schemes are based on approximating the second term on the right-hand side of Bellman's optimality principle (the expected value-to-go) by computationally tractable means. Although a wide range of approximate dynamic programming (ADP) methods have been developed by [47–49], a general systematic technique to provide performance guarantees for them has remained elusive. Ours is the first systematic approach to deriving performance bounds for general ADP methods in the stochastic setting.

In Chapter 7, we conclude this thesis and discuss some future research questions.

Chapter 2

Performance of Batched Greedy Strategy

In this chapter, we study the performance bounds in terms of the total batched curvature for the batched greedy strategy under general matroid and uniform matroid constraints. We also study the behavior of the bounds as functions of the batch size, by comparing the values of the total batched curvature for different batch sizes and investigating the monotoneity of the bounds. It is not our claim that we are proposing a new algorithm (the batched greedy strategy) or even that we are advocating the use of such an algorithm. Our contribution is to provide bounds on the performance of the batched greedy strategy, which we consider to be a rather natural extension of the greedy strategy. As we argue below, going from the case of batch size equal to 1 to the general case (batch size greater than 1) is highly nontrivial.

In [34], Conforti and Cornuéjols provided performance bounds for the greedy strategy in terms of the total curvature under general matroid constraints and uniform matroid constraints. It might be tempting to think that bounds for the batched case can be derived in a straightforward way from the results of batch size equal to 1 by lifting, which is to treat each batch-sized set of elements chosen by the batched greedy strategy as a single action, and then appeal to the results for the case of batch size equal to 1. However, it turns out that lifting does not work for a general batched greedy strategy (batch size greater than 1) for the following two reasons. First, the collection of sets created by satisfying the batched greedy strategy is not a matroid in general; we will demonstrate this by an example in Section 2.4. Second, the last step of the batched greedy strategy may select elements with a number less than the batch size, because the cardinality of the maximal set in the matroid may not be divisible by the batch size.

The batched greedy strategy requires an exponential number of evaluations of the objective function if using exhaustive search. When the batch size is equal to the cardinality of the maximal set in the matroid, the batched greedy strategy coincides with the optimal strategy. It might be tempting to expect that the batched strategy with batch size greater than 1 outperforms the usual greedy strategy, albeit at the expense of increasing computational complexity. Indeed, the Monte Carlo simulations performed in [51] for the maximum coverage problem show that the batched greedy strategy with batch size greater than 1 provides better approximation than the usual greedy strategy in many cases. However, it is also evident from their simulation that this is not always the case. In Section 2.5, we provide two examples of the maximum coverage problem where the usual greedy strategy performs better than the batched greedy strategy with batch size 2.

In Section 2.1, we first introduce some definitions and review the previous results. Then, we review Lemmas 1.1 and 1.2 from [52], which we will use to derive performance bounds for the batched greedy strategy under a uniform matroid constraint. In Section 2.2, we define the total batched curvature and then we provide a harmonic bound and an exponential bound for the batched greedy strategy under a general matroid constraint and a uniform matroid constraint, respectively, both in terms of the total batched curvature. We also prove that the batched curvature is nonincreasing in the batch size when the objective function is a polymatroid set function. This implies that the larger the batch size, the better the harmonic bound for a general matroid and when the batch size divides the rank of the uniform matroid, the larger the batch size, the better the exponential bound for a uniform matroid. In Section 2.3, we present a task scheduling problem and an adaptive sensing problem to demonstrate our results.

The results in this chapter were published in [50, 53].

2.1 Preliminaries

2.1.1 Polymatroid Set Functions and Curvature

The definitions and terminology in this paragraph are standard (see, e.g., [54–56]), but are included for completeness. Let X be a finite set, and $\mathcal{I}$ be a non-empty collection of subsets of X. Given a pair $(X, \mathcal{I})$, the collection $\mathcal{I}$ is said to be hereditary iff it satisfies property i below and has the augmentation property iff it satisfies property ii below:

i. For all $B \in \mathcal{I}$, any set $A \subseteq B$ is also in $\mathcal{I}$.

ii. For any $A, B \in \mathcal{I}$, if $|B| > |A|$, then there exists $j \in B \setminus A$ such that $A \cup \{j\} \in \mathcal{I}$.

The pair $(X, \mathcal{I})$ is called a matroid iff it satisfies both properties i and ii. The pair $(X, \mathcal{I})$ is called a uniform matroid iff $\mathcal{I} = \{S \subseteq X : |S| \leq K\}$ for a given K, called the rank of $(X, \mathcal{I})$.

Remark 2.1.1. *Three collections given as follows satisfy property i, property ii, and both, respectively.*

Let $X = \{a, b, c\}$, $\mathcal{I}_1 = \{\{a\}, \{b\}, \{a, c\}, \{c\}, \emptyset\}$, $\mathcal{I}_2 = \{\{a\}, \{a, b\}\}$, $\mathcal{I}_3 = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. *It is easy to check that* $\mathcal{I}_1$ *satisfies hereditary property but not augmentation,* $\mathcal{I}_2$ *satisfies augmentation but not hereditary property, and* $\mathcal{I}_3$ *satisfies both hereditary property and augmentation. Hence,* $(X, \mathcal{I}_1)$ *is an independence system and* $(X, \mathcal{I}_3)$ *is a matroid.*

Before we introduce the properties of function defined on sets, we would like to introduce some similar properties for functions defined on real numbers. Define a real function $f : \mathbb{R} \to \mathbb{R}$. The function f is said to monotone and submodular if it satisfies properties 1 and 2 below, respectively:

1. Monotone: $\forall x \leq y \in \mathbb{R}$, $f(x) \leq f(y)$.

2. Submodular: $\forall x \leq y \in \mathbb{R}$, $\forall z \in \mathbb{R}$, $f(x + z) - f(x) \geq f(y + z) - f(y)$

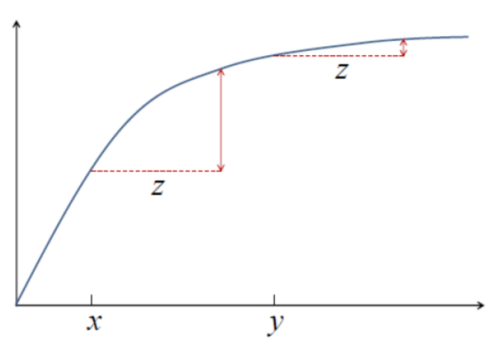

Figure 2.1: Characterization of submodularity

Remark 2.1.2. *We say that a function is monotone if it is nondecreasing.*

Remark 2.1.3. *From Figure 2.1, we can see that the function is a concave function and adding z to x gains more than adding z to y, which tells us that the additional value accruing by adding a number to a smaller number is larger than adding it to a bigger number. This is consistent with the inequality $f(x+z) - f(x) \geq f(y+z) - f(y)$ for $x \leq y$, so we say that submodularity is similar to concavity in some sense.*

Now look at some properties for functions defined on sets. Let 2^X denote the power set of X, and define a set function f: $2^X \longrightarrow \mathbb{R}$ The set function f is said to be monotone and submodular iff it satisfies properties (1) and (2) below, respectively:

(1) For any $A \subseteq B \subseteq X$, $f(A) \leq f(B)$.

(2) For any $A \subseteq B \subseteq X$ and $j \in X \setminus B$, $f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B)$.

A set function f: $2^X \longrightarrow \mathbb{R}$ is called a polymatroid set function iff it is monotone, submodular, and $f(\emptyset) = 0$, where $\emptyset$ denotes the empty set. The submodularity in property (2) means that the additional value accruing from an extra action decreases as the size of the input set increases, and is also called the diminishing-return property in economics. Submodularity implies that for any $A \subseteq B \subseteq X$ and $T \subseteq X \setminus B$,

$$f(A \cup T) - f(A) \geq f(B \cup T) - f(B). \tag{2.1}$$

For convenience, we denote the incremental value of adding a set T to the set $A \subseteq X$ as $\varrho_T(A) = f(A \cup T) - f(A)$ (following the notation of [34]).

The total curvature of a set function f is defined as [34]

$$c := \max_{j \in X^*} \left\{ 1 - \frac{\varrho_j(X \setminus \{j\})}{\varrho_j(\emptyset)} \right\},$$

where $X^* = \{j \in X : \varrho_j(\emptyset) \neq 0\}$. Note that $0 \leq c \leq 1$ when f is a polymatroid set function, and $c = 0$ if and only if f is additive, i.e., for any set $A \subseteq X$, $f(A) = \sum_{i \in A} f(\{i\})$. When $c = 0$, it is easy to check that the greedy strategy coincides with the optimal strategy. So in the rest of the paper, when we assume that f is a polymatroid set function, we only consider $c \in (0, 1]$.

2.1.2 Review of Previous Work

Before we review the previous work, we formulate the optimization problem formally as follows:

$$\text{maximize} \;\; f(M), \quad \text{subject to} \;\; M \in \mathcal{I}, \tag{2.2}$$

where $\mathcal{I}$ is a non-empty collection of subsets of a finite set X, and f is a real-valued set function defined on the power set 2^X of X.

For convenience, in the rest of the paper we will use k-batch greedy strategy to denote the batched greedy strategy with batch size k. So, the 1-batch greedy strategy denotes the usual greedy strategy.

Nemhauser et al. [5, 6] proved that, when f is a polymatroid set function, the 1-batch greedy strategy yields a $1/2$-approximation[2] for a general matroid and a $(1 - e^{-1})$-approximation for a uniform matroid. By introducing the total curvature c, Conforti and Cornuéjols [34] showed that, when f is a polymatroid set function, the 1-batch greedy strategy achieves a $1/(1 + c)$-approximation for a general matroid and a $(1 - e^{-c})/c$-approximation for a uniform matroid. For a polymatroid set function f, the total curvature c takes values on the interval $]0, 1]$. In this case, we have $1/(1 + c) \geq 1/2$ and $(1 - e^{-c})/c \geq (1 - e^{-1})$, which implies that the bounds $1/(1 + c)$ and $(1 - e^{-c})/c$ are stronger than the bounds $1/2$ and $(1 - e^{-1})$ in [5] and [6], respectively. Vondrák [35] proved that, when f is a polymatroid set function, the continuous greedy strategy gives a $(1 - e^{-c})/c$-approximation for any matroid. Sviridenko et al. [36] proved that, when f is a polymatroid set function, a modified continuous greedy strategy gives a $(1 - ce^{-1})$-approximation

[2]The term β-approximation means that $f(G)/f(O) \geq \beta$, where G and O denote a greedy solution and an optimal solution, respectively.

for any matroid, the first improvement over the greedy $(1-e^{-c})/c$-approximation of Conforti and Cornuéjols from [34].

Nemhauser et al. [6] proved that, when f is a polymatroid set function and $(X, \mathcal{I})$ is a uniform matroid of rank $K = kl + m$ (l and m are nonnegative integers and $0 < m \leq k$), the k-batch greedy strategy achieves a γ-approximation, where $\gamma = (1-(1-m/(k(l+1)))(1-1/(l+1))^l)$. Hausmann et al. [33] showed that, when f is a polymatroid set function and $(X, \mathcal{I})$ is an independence system, the k-batch greedy strategy achieves a $q(X, \mathcal{I})$-approximation, where $q(X, \mathcal{I})$ is the rank quotient defined in [33].

2.1.3 Performance Bounds in Terms of Total Curvature

In this section, we review two theorems from [34], which bound the performance of the 1-batch greedy strategy using the total curvature c for general matroid constraints and uniform matroid constraints. These bounds are special cases of the bounds we derive in Section 3.2 for $k = 1$.

We first define optimal and greedy solutions for problem (2.2) as follows:

Optimal solution: Consider problem (2.2) of finding a set that maximizes f under the constraint $M \in \mathcal{I}$. We call a solution of this problem an optimal solution and denote it by O, i.e.,

$$O \in \underset{M \in \mathcal{I}}{\operatorname{argmax}} f(M),$$

where argmax denotes the set of actions that maximize $f(\cdot)$.

1-batch greedy solution: A set $G = \{g_1, g_2, \ldots, g_k\}$ is called a 1-batch greedy solution if

$$g_1 \in \underset{g \in X}{\operatorname{argmax}} f(\{g\}),$$

and for $i = 2, \ldots, k$,

$$g_i \in \underset{g \in X}{\operatorname{argmax}} f(\{g_1, g_2, \ldots, g_{i-1}, g\}).$$

Theorem 2.1.1. [34] *Let $(X, \mathcal{I})$ be a matroid and $f\colon 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function with total curvature c. Then, any 1-batch greedy solution G satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{1+c},$$

where O is any optimal solution to problem (2.2).

When f is a polymatroid set function, we have $c \in (0, 1]$, and therefore $1/(1 + c) \in [1/2, 1)$. Theorem 2.1.1 applies to any matroid. This means that the bound $1/(1 + c)$ holds for a uniform matroid too. Theorem 2.1.2 below provides a tighter bound when $(X, \mathcal{I})$ is a uniform matroid.

Theorem 2.1.2. [34] *Let $(X, \mathcal{I})$ be a uniform matroid of rank K. Further, let $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function with total curvature c. Then, any 1-batch greedy solution G satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{c}\left(1 - \left(1 - \frac{c}{K}\right)^K\right) > \frac{1}{c}\left(1 - e^{-c}\right),$$

where O is any optimal solution to problem (2.2).

The function $(1-e^{-c})/c$ is a nonincreasing function of c, and therefore $(1-e^{-c})/c \in [1-e^{-1}, 1[$ when f is a polymatroid set function. Also it is easy to check that $(1 - e^{-c})/c \geq 1/(1 + c)$ for $c \in (0, 1]$, which implies that the bound $(1-e^{-c})/c$ is stronger than the bound $1/(1+c)$ in Theorem 2.1.1.

2.1.4 Properties of Submodular Functions

The following two lemmas from [52], stating some technical properties of submodular functions, will be useful to derive performance bounds for the k-batch greedy strategy under a uniform matroid constraint.

Lemma 2.1.3. [52] *Let $f\colon 2^X \longrightarrow \mathbb{R}$ be a submodular set function. Given $A, B \subseteq X$, let $\{M_1, \ldots, M_r\}$ be a collection of subsets of $B \setminus A$ such that each element of $B \setminus A$ appears in* *exactly p of these subsets. Then,*

$$\sum_{i=1}^{r} \varrho_{M_i}(A) \geq p\varrho_B(A).$$

Lemma 2.1.4. [52] *Let $f\colon 2^X \longrightarrow \mathbb{R}$ be a submodular set function. Given $A' \subseteq A \subseteq X$, let $\{T_1, \ldots, T_s\}$ be a collection of subsets of $A \setminus A'$ such that each element of $A \setminus A'$ appears in exactly q of these subsets. Then,*

$$\sum_{i=1}^{s} \varrho_{T_i}(A \setminus T_i) \leq q\varrho_{A\setminus A'}(A').$$

2.2 Main Results

In this section, first we define the k-batch greedy strategy and the total k-batch curvature c_k that will be used for deriving harmonic and exponential bounds. Then we derive performance bounds for the k-batch greedy strategy in terms of c_k under general matroid constraints and under uniform matroid constraints. Moreover, we study the behavior of the bounds as functions of the batch size k.

2.2.1 k-Batch Greedy Strategy

We write the cardinality of the maximal set in $\mathcal{I}$ as $K = kl + m$, where l, m are nonnegative integers and $0 < m \leq k$. Note that m is not necessarily the remainder of K/k, because m could be equal to k. This happens when k divides K. The k-batch greedy strategy is as follows:

Step 1: Let $S^0 = \emptyset$ and $t = 0$.

Step 2: Select $J_{t+1} \subseteq X \setminus S^t$ for which $|J_{t+1}| = k$, $S^t \cup J_{t+1} \in \mathcal{I}$, and

$$f(S^t \cup J_{t+1}) = \max_{J\subseteq X\setminus S^t \text{ and } |J|=k} f(S^t \cup J);$$

then set $S^{t+1} = S^t \cup J_{t+1}$.

Step 3: If $t + 1 < l$, set $t = t + 1$, and repeat Step 2.

Step 4: If $t + 1 = l$, select $J_{l+1} \subseteq X \setminus S^l$ such that $|J_{l+1}| = m$, $S^l \cup J_{l+1} \in \mathcal{I}$, and

$$f(S^l \cup J_{l+1}) = \max_{J \subseteq X \setminus S^l \text{ and } |J|=m} f(S^l \cup J).$$

Step 5: Return the set $S = S^l \cup J_{l+1}$ and terminate.

Any set generated by the above procedure is called a k-batch greedy solution.

The difference between a k-batch greedy strategy for a general matroid and that for a uniform matroid is that at each step t ($0 \leq t \leq l$), we have to check whether $J_{t+1} \subseteq X \setminus S^t$ satisfies $S^t \cup J_{t+1} \in \mathcal{I}$ for a general matroid while $S^t \cup J_{t+1} \in \mathcal{I}$ always holds for a uniform matroid.

2.2.2 Performance Bounds in Terms of Total k-Batch Curvature

Similar to the definition of the total curvature c in [34], we define the total k-batch curvature c_k for a given k as

$$c_k := \max_{I \in \hat{X}} \left\{ 1 - \frac{\varrho_I(X \setminus I)}{\varrho_I(\emptyset)} \right\}, \tag{2.3}$$

where $\hat{X} = \{I \subseteq X : \varrho_I(\emptyset) \neq 0 \text{ and } |I| = k\}$.

The following proposition will be applied to derive our bounds in terms of c_k for both general matroid constraints and uniform matroid constraints.

Proposition 2.2.1. *If $f : 2^X \longrightarrow \mathbb{R}$ is a submodular set function, $A, B \subseteq X$, and $\{M_1, \ldots, M_r\}$ is a partition of $B \setminus A$, then*

$$f(A \cup B) \leq f(A) + \sum_{i : M_i \subseteq B \setminus A} \varrho_{M_i}(A). \tag{2.4}$$

Proof. By the assumption that $\{M_1, \ldots, M_r\}$ is a partition of $B \setminus A$ and by submodularity (see inequality (2.1)), we have

$$
\begin{aligned}
f(A \cup B) - f(A) &= f(A \cup \bigcup_{j=1}^{r} M_j) - f(A) \\
&= \sum_{i=1}^{r} \varrho_{M_i}(A \cup \bigcup_{j=1}^{i-1} M_j) \\
&\leq \sum_{i: M_i \subseteq B \setminus A} \varrho_{M_i}(A),
\end{aligned}
$$

which implies inequality (2.4). □

The following proposition in terms of the total k-batch curvature c_k will be applied to derive our bounds under general matroid constraints.

Proposition 2.2.2. *Let $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function. Given a set $B \subseteq X$, a sequence of t $(t > 0)$ sets $A^i = \bigcup_{j=1}^{i} I_j$ with $I_j \subseteq X$ and $|I_j| = k$ for $1 \leq j \leq t$, and a partition $\{M_1, \ldots, M_r\}$ of $B \setminus A^t$, we have*

$$
f(B) \leq c_k \sum_{i: I_i \subseteq A^t \setminus B} \varrho_{I_i}(A^{i-1}) + \sum_{i: I_i \subseteq B \cap A^t} \varrho_{I_i}(A^{i-1}) + \sum_{i: M_i \subseteq B \setminus A^t} \varrho_{M_i}(A^t). \tag{2.5}
$$

Proof. By the definition of A^t, we write

$$
f(A^t \cup B) - f(B) = \sum_{i=1}^{t} \varrho_{I_i}(B \cup A^{i-1}) = \sum_{i: I_i \subseteq A^t \setminus B} \varrho_{I_i}(B \cup A^{i-1}).
$$

By submodularity (see inequality (2.1)), we have

$$
\varrho_{I_i}(B \cup A^{i-1}) \geq \varrho_{I_i}(X \setminus I_i) \tag{2.6}
$$

and

$$
\varrho_{I_i}(\emptyset) \geq \varrho_{I_i}(A^{i-1}) \tag{2.7}
$$

for $1 \leq i \leq t$. By the definition of the total k-batch curvature c_k, we have

$$1-\frac{\varrho_{I_i}(X \setminus I_i)}{\varrho_{I_i}(\emptyset)} \le c_k$$

for $1 \le i \le t$, which implies that

$$\varrho_{I_i}(X \setminus I_i) \ge (1-c_k)\varrho_{I_i}(\emptyset).$$

Combining the above inequality with (2.6) and (2.7), we have

$$\varrho_{I_i}(B \cup A^{i-1}) \ge \varrho_{I_i}(X \setminus I_i) \ge (1-c_k)\varrho_{I_i}(\emptyset) \ge (1-c_k)\varrho_{I_i}(A^{i-1})$$

for $1 \le i \le t$. Using the above inequality, we have

$$\begin{aligned} f(A^t \cup B) - f(B) &= \sum_{i:I_i \subseteq A^t \setminus B} \varrho_{I_i}(B \cup A^{i-1}) \\ &\ge (1-c_k) \sum_{i:I_i \subseteq A^t \setminus B} \varrho_{I_i}(A^{i-1}). \end{aligned} \tag{2.8}$$

By Proposition 2.2.1, we have

$$f(A^t \cup B) \le f(A^t) + \sum_{i:M_i \subseteq B \setminus A^t} \varrho_{M_i}(A^t). \tag{2.9}$$

Combining inequalities (2.8) and (2.9) results in

$$f(B) \le f(A^t) + \sum_{i:M_i \subseteq B \setminus A^t} \varrho_{M_i}(A^t) - (1-c_k) \sum_{i:I_i \subseteq A^t \setminus B} \varrho_{I_i}(A^{i-1}).$$

Substituting $f(A^t)$ into the above inequality by the identity

$$f(A^t) = \sum_{i:I_i \subseteq A^t \setminus B} \varrho_{I_i}(A^{i-1}) + \sum_{i:I_i \subseteq B \cap A^t} \varrho_{I_i}(A^{i-1}),$$

we get inequality (2.5). $\square$

Recall that in Section 2.2.1, we defined J_i as the set selected by the k-batch greedy strategy at stage i and $S^i = \bigcup_{j=1}^{i} J_j$ as the set selected by the k-batch greedy strategy for the first i stages, where $1 \leq i \leq l+1$, $|J_i| = k$ for $1 \leq i \leq l$, $|J_{l+1}| = m$, and $K = kl + m$ with $l \geq 0$ and $0 < m \leq k$ being integers. When the pair $(X, \mathcal{I})$ is a matroid, by the augmentation property of a matroid and the previous assumption that the maximal cardinality of $\mathcal{I}$ is K, we have that any optimal solution can be augmented to a set of length K. Assume that $O = \{o_1, \ldots, o_K\}$ is an optimal solution to problem (2.2). Let $S = S^{l+1}$ be a k-batch greedy solution. We now state and prove the following lemma, which will be used to derive the harmonic bound for general matroid constraints in Theorem 2.2.4.

Lemma 2.2.3. *Let S be a k-batch greedy solution and $O = \{o_1, \ldots, o_K\}$ be an optimal solution. Then the following statements hold:*

a. There exists a partition $\{J'_i\}_{i=1}^{l+1}$ of O with $|J'_i| = k$ for $1 \leq i \leq l$ and $|J'_{l+1}| = m$ such that $\varrho_{J'_i}(S^{i-1}) \leq \varrho_{J_i}(S^{i-1})$. Furthermore, if $J'_i \subseteq O \cap S$, then $J'_i = J_i$.

b. If $J'_i \subseteq O \setminus S^l$ for $1 \leq i \leq l$, then $J_i \subseteq S^l \setminus O$.

Proof. We begin by proving a. First, we prove that there exists $J'_{l+1} \subseteq O \backslash S^l$ such that $S^l \cup J'_{l+1} \in \mathcal{I}$ and $\varrho_{J'_{l+1}}(S^l) \leq \varrho_{J_{l+1}}(S^l)$. By definition, $|O| = K$ and $S^l = kl = K - m$. Using the augmentation property, there exists one element $o_{i_1} \in O \setminus S^l$ such that $S^l \cup \{o_{i_1}\} \in \mathcal{I}$. Consider $S^l \cup \{o_{i_1}\}$ and O. Using the augmentation property again, there exists one element $o_{i_2} \in O \setminus S^l \setminus \{o_{i_1}\}$ such that $S^l \cup \{o_{i_1}, o_{i_2}\} \in \mathcal{I}$. Using the augmentation property $(m-2)$ more times, we have that there exists $J'_{l+1} = \{o_{i_1}, \ldots, o_{i_m}\} \subseteq O \setminus S^l$ such that $S^l \cup J'_{l+1} \in \mathcal{I}$. By the k-batch greedy strategy, we have $\varrho_{J'_{l+1}}(S^l) \leq \varrho_{J_{l+1}}(S^l)$. If $J_{l+1} \subseteq O$, we can set $J'_{l+1} = J_{l+1}$.

Then similar to the proof in [6], we will prove statement a by backward induction on i for $i = l, l-1, \ldots, 1$. Assume that J'_i satisfies the inequality $\varrho_{J'_i}(S^{i-1}) \leq \varrho_{J_i}(S^{i-1})$ for $i > j$, and let $O^j = O \backslash \bigcup_{i>j} J'_i$. Consider the sets S^{j-1} and O^j. By definition, $|S^{j-1}| = (j-1)k$ and $|O^j| = jk$. Using the augmentation property, we have that there exists one element $o_{j_1} \in O^j \setminus S^{j-1}$ such that $S^{j-1} \cup \{o_{j_1}\} \in \mathcal{I}$. Next consider $S^{j-1} \cup \{o_{j_1}\}$ and O^j. Using the augmentation property again, there exists one element $o_{j_2} \in O^j \setminus S^{j-1} \setminus \{o_{j_1}\}$ such that $S^{j-1} \cup \{o_{j_1}, o_{j_2}\} \in \mathcal{I}$. Similar to the process above, using the augmentation property $(k-2)$ more times, finally we have that there exists $J'_j = \{o_{j_1}, \dots, o_{j_k}\} \subseteq O^j \setminus S^{j-1}$ such that $S^{j-1} \cup J'_j \in \mathcal{I}$. By the k-batch greedy strategy, we have that $\varrho_{J'_j}(S^{j-1}) \leq \varrho_{J_j}(S^{j-1})$. Furthermore, if $J_j \subseteq O^j$, we can set $J'_j = J_j$. This completes the proof of statement a.

Now we prove statement b by contradiction. Consider the negation of statement b, i.e., if $J'_i \subseteq O \setminus S^l$ for $1 \leq i \leq l$, then $J_i \subseteq O$. By the argument in the second paragraph of the proof of statement a, we have that if $J_i \subseteq O$ for $1 \leq i \leq l$, then $J_i = J'_i$. By the assumption that $J'_i \subseteq O \setminus S^l$ for $1 \leq i \leq l$, we have $J_i \subseteq O \setminus S^l$ for $1 \leq i \leq l$, which contradicts the fact that $J_i \subseteq S^l$ for $1 \leq i \leq l$. This completes the proof of statement b. □

The following theorem presents our performance bound in terms of the total k-batch curvature c_k for the k-batch greedy strategy under a general matroid.

Theorem 2.2.4. *Let $(X, \mathcal{I})$ be a general matroid and $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function. Then, any k-batch greedy solution S satisfies*

$$\frac{f(S)}{f(O)} \geq \frac{1}{1 + c_k}. \tag{2.10}$$

Proof. Let $\{P_{i_1}, \dots, P_{i_r}\}$ be a partition of $O \setminus S^l$ satisfying that $P_{i_j} \subseteq J'_{i_j}$ for $1 \leq j \leq r$. The way to find $\{P_{i_1}, \dots, P_{i_r}\}$ is as follows: first list all of the actions in $O \setminus S^l$, then let P_i be its subset consisting of actions belonging to J'_i, i.e., $P_i = (O \setminus S^l) \cap J'_i$. Finally, extract the nonempty sets from $\{P_i\}_{i=1}^{l+1}$ as $\{P_{i_1}, \dots, P_{i_r}\}$.

Recall that $S^i = \cup_{j=1}^{i} J_j$ for $1 \leq i \leq l$ as defined in Section 2.2.1. Then using Proposition 2.2.2, with $A^t = S^l$ and $S = O$ results in

$$
\begin{aligned}
f(O) &\leq c_k \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{i:P_i \subseteq O \setminus S^l} \varrho_{P_i}(S^l) \\
&= c_k \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{\substack{i:P_i \subseteq O \setminus S^l \\ i \neq l+1}} \varrho_{P_i}(S^l) + \varrho_{P_{l+1}}(S^l). \qquad (2.11)
\end{aligned}
$$

By the monotoneity of the set function f and because $P_{i_j} \subseteq J'_{i_j}$ for $1 \leq j \leq r$, we have

$$
\varrho_{P_{i_j}}(S^l) \leq \varrho_{J'_{i_j}}(S^l) \qquad (2.12)
$$

for $1 \leq j \leq r$. Based on the fact that $J'_i \subseteq O$ for $1 \leq i \leq l+1$, and because $P_{i_j} \subseteq J'_{i_j}$ and $P_{i_j} \subseteq O \setminus S^l$, we have

$$
J'_{i_j} \subseteq O \setminus S^l. \qquad (2.13)
$$

Combining (2.11)-(2.13) results in

$$
f(O) \leq c_k \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{\substack{i:J'_i \subseteq O \setminus S^l \\ i \neq l+1}} \varrho_{J'_i}(S^l) + \varrho_{J'_{l+1}}(S^l). \qquad (2.14)
$$

By submodularity (see inequality (2.1)), we have

$$
\varrho_{J'_i}(S^l) \leq \varrho_{J'_i}(S^{i-1}) \qquad (2.15)
$$

for $1 \leq i \leq l+1$. By statement a in Lemma 2.2.3, we have

$$
\varrho_{J'_i}(S^{i-1}) \leq \varrho_{J_i}(S^{i-1}) \qquad (2.16)
$$

for $1 \leq i \leq l+1$. By combining inequalities (2.15) and (2.16), we have

$$
\varrho_{J'_i}(S^l) \leq \varrho_{J_i}(S^{i-1}), \qquad (2.17)
$$

for $1 \leq i \leq l+1$. Combining inequalities (2.14) and (2.17) results in

$$f(O) \leq c_k \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{\substack{i:J_i' \subseteq O \setminus S^l \\ i \neq l+1}} \varrho_{J_i}(S^{i-1}) + \varrho_{J_{l+1}}(S^l). \tag{2.18}$$

By statement b in Lemma 2.2.3, and inequality (2.18), we have

$$f(O) \leq c_k \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) + \varrho_{J_{l+1}}(S^l). \tag{2.19}$$

Because

$$\sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) \leq f(S^l),$$

$$\sum_{i:J_i \subseteq O \cap S^l} \varrho_{J_i}(S^{i-1}) + \sum_{i:J_i \subseteq S^l \setminus O} \varrho_{J_i}(S^{i-1}) = f(S^l),$$

and

$$\varrho_{J_{l+1}}(S^l) = f(S) - f(S^l),$$

we can use inequality (2.19) to write

$$f(O) \leq c_k f(S^l) + f(S^l) + f(S) - f(S^l) \leq (c_k + 1) f(S),$$

which implies that $f(S)/f(O) \geq \frac{1}{1+c_k}$. □

Remark 2.2.1. *For $k = 1$, the harmonic bound for a general matroid becomes the bound in Theorem 2.1.1.*

Remark 2.2.2. *The function $g(x) = 1/(1+x)$ is nonincreasing in x on the interval $(0, 1]$.*

Remark 2.2.3. *The harmonic bound $1/(1+c_k)$ for the k-batch greedy strategy holds for any matroid. For the special case of a uniform matroid, we will give a different (exponential) bound in Theorem 2.2.6 below. We will also show that this exponential bound is better than the harmonic bound when k divides the rank of the uniform matroid K.*

In Theorem 2.2.6 below, we provide an exponential bound for the k-batch greedy strategy in the case of uniform matroids. The special case when $c_k = 1$ was derived in [6]. Our result here is more general, and the method used in our proof is different from that of [6]. The new proof here is of particular interest because the technique here is not akin to that used in the case of general matroids in Theorem 2.2.4 and also was not considered in [6]. Before stating the theorem, we first present a proposition that will be used in proving Theorem 2.2.6.

Choose a set $J^* \subseteq X \setminus S^l$ with $|J^*| = k$ so as to maximize $f(S^l \cup J^*) - f(S^l)$. Write $\varrho_{J_{l+1}}(S^l) = f(S^{l+1}) - f(S^l)$ and $\varrho_{J^*}(S^l) = f(S^l \cup J^*) - f(S^l)$. We have the following proposition.

Proposition 2.2.5. *Let $f : 2^X \longrightarrow \mathbb{R}$ be a submodular set function. Then when $(X, \mathcal{I})$ is a uniform matroid, we have $\varrho_{J_{l+1}}(S^l) \geq \frac{m}{k} \varrho_{J^*}(S^l)$.*

Proof. Let $\{M_1, \ldots, M_r\}$, where

$$r = \binom{k}{m},$$

be the collection of all the subsets of J^* with cardinality m. Then, each element of J^* appears in exactly p of these subsets, where

$$p = \binom{k-1}{m-1}.$$

Using Lemma 2.1.3 with $A = S^l, B = J^*$ and $B \setminus A = J^*$, we have

$$\sum_{i=1}^{r} \varrho_{M_i}(S^l) \geq p \varrho_{J^*}(S^l). \tag{2.20}$$

Because $|S^l \cup M_i| = kl + m = K$, by the definition of the uniform matroid $(X, \mathcal{I})$, we have $S^l \cup M_i \in \mathcal{I}$. By the definition of the k-batch greedy strategy and the monotoneity of the set function f, we have

$$\varrho_{J_{l+1}}(S^l) \geq \varrho_{M_i}(S^l), \tag{2.21}$$

which implies that

$$r \varrho_{J_{l+1}}(S^l) \geq \sum_{i=1}^{r} \varrho_{M_i}(S^l). \tag{2.22}$$

Combining (2.22) and (2.20), we have

$$\varrho_{J_{l+1}}(S^l) \geq \frac{1}{r}\sum_{i=1}^{r}\varrho_{M_i}(S^l) \geq \frac{p}{r}\varrho_{J^*}(S^l) \geq \frac{m}{k}\varrho_{J^*}(S^l),$$

which implies that $\varrho_{J_{l+1}}(S^l) \geq \frac{m}{k}\varrho_{J^*}(S^l)$. $\square$

Remark 2.2.4. *The reason we require $(X, \mathcal{I})$ to be a uniform matroid is that this result does not necessarily hold for a general matroid, because $S^l \cup M_i \in \mathcal{I}$ is not guaranteed for a general matroid, and in consequence inequality (2.21) does not necessarily hold.*

Theorem 2.2.6. *Let $(X, \mathcal{I})$ be a uniform matroid and $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function. Then, any k-batch greedy solution S satisfies*

$$\frac{f(S)}{f(O)} \geq \frac{1}{c_k}\left(1 - \left(1 - \frac{c_k}{l+1}\frac{m}{k}\right)\left(1 - \frac{c_k}{l+1}\right)^l\right). \tag{2.23}$$

Proof. Recall again that J_i is the set selected at stage i by the k-batch greedy strategy, $S^i = \cup_{j=1}^{i} J_j$ for $1 \leq i \leq l$, and $S^0 = \emptyset$ as defined in Section 3.1. Also recall that we defined J^* as the set that maximizes $f(S^l \cup J^*) - f(S^l)$ with $J^* \subseteq X \setminus S^l$ and $|J^*| = k$.

Let $\{P_{i,1}, \ldots, P_{i,r_i}\}$ be a partition of $O \setminus S^i$ satisfying $P_{i,j} \subseteq J'_{i,j}$ for $1 \leq j \leq r_i$. Finding $\{P_{i,1}, \ldots, P_{i,r_i}\}$ for each i is similar to finding $\{P_{i_1}, \ldots, P_{i_r}\}$ which was given in the proof of Theorem 2.2.4. Letting $B = O$ and $A = S^i$ ($0 \leq i \leq l$) in Proposition 2.2.1, we have

$$f(O \cup S^i) \leq f(S^i) + \sum_{j: P_{i,j} \subseteq O \setminus S^i} \varrho_{P_{i,j}}(S^i). \tag{2.24}$$

By the monotoneity of the set function f and because $P_{i,j} \subseteq J'_{i,j}$ for $1 \leq j \leq r_i$, we have

$$\varrho_{P_{i,j}}(S^i) \leq \varrho_{J'_{i,j}}(S^i). \tag{2.25}$$

Based on the fact that $J'_{i,j} \subseteq O$ and because $P_{i,j} \subseteq O \setminus S^i$ and $P_{i,j} \subseteq J'_{i,j}$, we have

$$J'_{i,j} \subseteq O \setminus S^i. \tag{2.26}$$

Combining (2.24)-(2.26) results in

$$f(O \cup S^i) \leq f(S^i) + \sum_{j: J'_{i,j} \subseteq O \setminus S^i} \varrho_{J'_{i,j}}(S^i). \tag{2.27}$$

For $0 \leq i \leq l-1$, we have $|S^i \cup J'_{i,j}| \leq K$, which implies that $S^i \cup J'_{i,j} \in \mathcal{I}$ always holds. So for any given i ($0 \leq i \leq l-1$), by the definition of the k-batch greedy strategy, we have

$$\varrho_{J'_{i,j}}(S^i) \leq \varrho_{J_{i+1}}(S^i) \tag{2.28}$$

for any $J'_{i,j} \subseteq O \setminus S^i$. Now consider $i = l$. For any $J'_{l,j} \subseteq O \setminus S^l$ with $|J'_{l,j}| = k$, by the definition of J^* before Proposition 2.2.5, we have

$$\varrho_{J'_{l,j}}(S^l) \leq \varrho_{J^*}(S^l).$$

By the definition of J^* and the monotoneity of the set function f, we have

$$\varrho_{J'_{l+1}}(S^l) \leq \varrho_{J^*}(S^l).$$

Combining the two inequalities above, we have for any $J'_{l,j} \subseteq O \setminus S^l$,

$$\varrho_{J'_{l,j}}(S^l) \leq \varrho_{J^*}(S^l). \tag{2.29}$$

By inequalities (2.28) and (2.29), for any given i ($0 \leq i \leq l$), we have

$$\varrho_{J'_{i,j}}(S^i) \leq \varrho_{L_{i+1}}(S^i) \tag{2.30}$$

for any $J'_{i,j} \subseteq O \setminus S^i$, where

$$L_{i+1} = \begin{cases} J_{i+1}, & 0 \leq i \leq l-1, \\ J^*, & i = l. \end{cases}$$

By inequalities (2.27) and (2.30), we have

$$f(O \cup S^i) \leq f(S^i) + \sum_{j: J'_{i,j} \subseteq O \setminus S^i} \varrho_{L_{i+1}}(S^i),$$

which implies that

$$f(O \cup S^i) \leq f(S^i) + (l+1)\varrho_{L_{i+1}}(S^i). \tag{2.31}$$

Setting $i = 0$ in inequality (2.31), recalling that $S^0 = \emptyset$, and because $S^1 = J_1$ by definition, we have

$$f(S^1) \geq \frac{1}{l+1} f(O). \tag{2.32}$$

For $1 \leq i \leq l$, we write

$$\frac{f(O \cup S^i) - f(O)}{f(S^i)} = \frac{\sum_{j=0}^{i-1} f(O \cup S^j \cup J_{j+1}) - f(O \cup S^j)}{\sum_{j=0}^{i-1} f(S^j \cup J_{j+1}) - f(S^j)}. \tag{2.33}$$

By submodularity (see (2.1)), we have

$$f(O \cup S^j \cup J_{j+1}) - f(O \cup S^j) \geq f(X) - f(X \setminus J_{j+1}) \tag{2.34}$$

and

$$f(S^j \cup J_{j+1}) - f(S^j) \leq f(J_{j+1}) - f(\emptyset) \tag{2.35}$$

for $0 \leq j \leq i-1$. By the definition of the total k-batch curvature, we have

$$\frac{f(X) - f(X \setminus J_{j+1})}{f(J_{j+1}) - f(\emptyset)} \geq 1 - c_k \tag{2.36}$$

for $0 \leq j \leq i-1$. Combining inequalities (2.33)-(2.35) results in

$$\frac{f(O \cup S^i) - f(O)}{f(S^i)} \geq \frac{\sum_{j=0}^{i-1} f(X) - f(X \setminus J_{j+1})}{\sum_{j=0}^{i-1} f(J_{j+1}) - f(\emptyset)} \geq 1 - c_k.$$

This in turn implies that

$$f(O) + (1 - c_k) f(S^i) \leq f(O \cup S^i).$$

Combining the above inequality and (2.31), we have

$$f(S^i \cup L_{i+1}) \geq \frac{1}{l+1} f(O) + \left(1 - \frac{c_k}{l+1}\right) f(S^i) \tag{2.37}$$

for $1 \leq i \leq l$. By inequality (2.32) and successive application of inequality (2.37) for $i = 1, \ldots, l$, we have

$$\begin{aligned} f(S^l) &\geq \frac{1}{l+1} f(O) + \left(1 - \frac{c_k}{l+1}\right) f(S^{l-1}) \\ &\geq \frac{1}{l+1} f(O) \sum_{i=0}^{l-1} \left(1 - \frac{c_k}{l+1}\right)^i \\ &= \frac{1}{c_k} \left(1 - \left(1 - \frac{c_k}{l+1}\right)^l\right) f(O), \end{aligned} \tag{2.38}$$

and

$$\begin{aligned} f(S^l \cup J^*) &\geq \frac{1}{l+1} f(O) + \left(1 - \frac{c_k}{l+1}\right) f(S^l) \\ &\geq \frac{1}{l+1} f(O) \sum_{i=0}^{l} \left(1 - \frac{c_k}{l+1}\right)^i \\ &\geq \frac{1}{c_k} \left(1 - \left(1 - \frac{c_k}{l+1}\right)^{l+1}\right) f(O). \end{aligned} \tag{2.39}$$

Using Proposition 2.2.5 and combining inequalities (2.38) and (2.39), we have

$$
\begin{aligned}
f(S) &\geq \frac{m}{k} f(S^l \cup J^*) + \left(1 - \frac{m}{k}\right) f(S^l) \\
&\geq \frac{m}{k}\frac{1}{c_k}\left(1 - \left(1 - \frac{c_k}{l+1}\right)^{l+1}\right) f(O) + \left(1 - \frac{m}{k}\right)\frac{1}{c_k}\left(1 - \left(1 - \frac{c_k}{l+1}\right)^{l}\right) f(O) \\
&= \frac{1}{c_k}\left(1 - \left(1 - \frac{c_k}{l+1}\frac{m}{k}\right)\left(1 - \frac{c_k}{l+1}\right)^{l}\right) f(O),
\end{aligned}
$$

which implies (2.23). □

Remark 2.2.5. *For $k = 1$, the exponential bound for a uniform matroid becomes the bound in Theorem 2.1.2.*

Remark 2.2.6. *The exponential bound for a uniform matroid becomes*

$$
1 - \left(1 - \frac{1}{l+1}\frac{m}{k}\right)\left(1 - \frac{1}{l+1}\right)^{l}
$$

for $c_k = 1$, which is the bound in [6].

Remark 2.2.7. *When $m = k$, i.e., when k divides the cardinality K, the exponential bound for a uniform matroid becomes*

$$
\frac{1}{c_k}\left(1 - \left(1 - \frac{c_k}{l+1}\right)^{l+1}\right),
$$

which is the bound in [50].

Remark 2.2.8. *Let $g(x, y) = (1 - (1 - x/y)^y)/x$. The function $g(x, y)$ is nonincreasing in x on the interval $(0, 1]$ for any positive integer y. Also, $g(x, y)$ is nonincreasing in y when x is a constant on the interval $(0, 1]$.*

Remark 2.2.9. *Even if the total curvature c_k is monotone in k, the exponential bound for a uniform matroid is not necessarily monotone. But under the condition that k divides K, it is monotone. To be specific, if k divides K, then $K = k(l+1)$ for some positive integer l. Thus, as k increases, $l+1$ decreases, and if c_k decreases, then, we have that $(1-(1-c_k/(l+1))^{l+1})/c_k$ is nondecreasing in k based on the previous remark.*

Remark 2.2.10. *When $m = k$, the exponential bound is tight, as shown in [6]. Moreover, for this case, the exponential bound $(1-(1-c_k/(l+1)\cdot m/k)(1-c_k/(l+1))^l)/c_k$ is better than the harmonic bound $1/(1+c_k)$ because*

$$\frac{1}{c_k}\left(1-\left(1-\frac{c_k}{l+1}\right)^{l+1}\right) > \frac{1-e^{-c_k}}{c_k} \geq \frac{1}{1+c_k}.$$

However, if k does not divide K, the exponential bound might be worse than the harmonic bound. For example, when $K = 100, k = 80$, and $c_k = 0.6$, the exponential bound is 0.5875, which is worse than the harmonic bound 0.6250.

Remark 2.2.11. *The monotoneity of $1/(1+c_k)$ implies that the k-batch greedy strategy has a better harmonic bound than the 1-batch greedy strategy if $c_k \leq c$. The monotoneity of $(1-(1-c_k/(l+1))^{l+1})/c_k$ implies that the k-batch (k divides K) greedy strategy has a better exponential bound than the 1-batch greedy strategy if $c_k \leq c$.*

The following theorem establishes that indeed $c_k \leq c$.

Theorem 2.2.7. *Let $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function with total curvature c and total k-batch curvatures $\{c_k\}_{k=1}^K$. Then, $c_k \leq c$ for $1 \leq k \leq K$.*

Proof. By the definition of the total k-batch curvature c_k, we have

$$c_k = \max_{I\in\hat{X}}\left\{1-\frac{\varrho_J(X\setminus I)}{\varrho_I(\emptyset)}\right\} = 1-\min_{I\in\hat{X}}\left\{\frac{\sum_{j=1}^{k}\varrho_{i_j}(X\setminus I_j)}{\sum_{j=1}^{k}\varrho_{i_j}(I_{j-1})}\right\},$$

where $I = \{i_1,\dots,i_k\}$ and $I_j = \{i_1,\dots,i_j\}$ for $1 \leq j \leq k$.

By submodularity (see (2.1)), we have

$$\varrho_{i_j}(X\setminus I_j) \geq \varrho_{i_j}(X\setminus\{i_j\}) \text{ and } \varrho_{i_j}(I_{j-1}) \leq \varrho_{i_j}(\emptyset)$$

for $1 \leq j \leq k$, which imply that

$$\frac{\sum_{j=1}^{k} \varrho_{i_j}(X \setminus I_j)}{\sum_{j=1}^{k} \varrho_{i_j}(I_{j-1})} \geq \frac{\sum_{j=1}^{k} \varrho_{i_j}(X \setminus \{i_j\})}{\sum_{j=1}^{k} \varrho_{i_j}(\emptyset)}.$$

Therefore, we have

$$c_k \leq 1 - \min_{I_k \in \hat{X}} \left\{ \frac{\sum_{j=1}^{k} \varrho_{i_j}(X \setminus \{i_j\})}{\sum_{j=1}^{k} \varrho_{i_j}(\emptyset)} \right\}. \tag{2.40}$$

By the definition of c and the fact that f is a polymatroid set function, we have $\varrho_{i_j}(X \setminus \{i_j\}) \geq (1-c)\varrho_{i_j}(\emptyset)$ for $1 \leq i \leq k$. Combining this inequality and (2.40), we have $c_k \leq 1-(1-c) = c$. $\square$

One would expect the following generalization of Theorem 2.2.7 to hold: if $k_2 \geq k_1$, then $c_{k_2} \leq c_{k_1}$. In the case of general matroid constraints, this conclusion implies that the bound is nondecreasing in k. In the case of uniform matroid constraints, monotoneity of the bound holds under the condition that k divides K. We now state and prove the following theorem on the monotoneity of c_k, using Lemmas 2.1.3 and 2.1.4 (Lemmas 1.1 and 1.2 in [52]).

Theorem 2.2.8. *Let $f : 2^X \longrightarrow \mathbb{R}$ be a polymatroid set function with k-batch curvatures $\{c_k\}_{k=1}^{K}$. Then, $c_{k_2} \leq c_{k_1}$ whenever $k_2 \geq k_1$.*

Proof. Let $J \subseteq X$ be a set with cardinality k_2 satisfying $f(J) > 0$. Let $\{M_1, \ldots, M_s\}$ be the collection of all the subsets of J with cardinality k_1 ($k_1 \leq k_2$), where

$$s = \binom{k_2}{k_1}.$$

Then, each element of J appears in exactly q of the subsets $\{M_1, \ldots, M_s\}$, where

$$q = \binom{k_2 - 1}{k_1 - 1}.$$

Using Lemma 2.1.4 with $A = X$, $A' = X \setminus J$, and $A \setminus A' = J$, we have

$$\sum_{i=1}^{s} \varrho_{M_i}(X \setminus M_i) \leq q\varrho_J(X \setminus J),$$

which implies that

$$\varrho_J(X \setminus J) \geq \frac{1}{q} \sum_{i=1}^{s} \varrho_{M_i}(X \setminus M_i). \tag{2.41}$$

Based on the fact that $\{M_1, \ldots, M_s\}$ is the collection of all the subsets of J with cardinality k_1 and that each element of J appears in exactly q of these subsets, using Lemma 2.1.3 with $B = J$ and $A = \emptyset$, we have

$$\sum_{i=1}^{s} \varrho_{M_i}(\emptyset) \geq q\varrho_J(\emptyset),$$

which implies that

$$\varrho_J(\emptyset) \leq \frac{1}{q} \sum_{i=1}^{s} \varrho_{M_i}(\emptyset). \tag{2.42}$$

Combining inequalities (2.41) and (2.42) results in

$$\frac{\varrho_J(X \setminus J)}{\varrho_J(\emptyset)} \geq \frac{\frac{1}{q}\sum_{i=1}^{s} \varrho_{M_i}(X \setminus M_i)}{\frac{1}{q}\sum_{i=1}^{s} \varrho_{M_i}(\emptyset)} = \frac{\sum_{i=1}^{s} \varrho_{M_i}(X \setminus M_i)}{\sum_{i=1}^{s} \varrho_{M_i}(\emptyset)}. \tag{2.43}$$

Recall the definition of the total k-batch curvature c_k in (2.3). Because $|M_i| = k_1$ for $1 \leq i \leq s$ and f is a polymatroid set function, we have

$$\varrho_{M_i}(X \setminus M_i) \geq (1 - c_{k_1})\varrho_{M_i}(\emptyset) \tag{2.44}$$

for $1 \leq i \leq s$. Combining inequalities (2.43) and (2.44) results in

$$\frac{\varrho_J(X \setminus J)}{\varrho_J(\emptyset)} \geq \frac{(1 - c_{k_1})\sum_{i=1}^{s} \varrho_{M_i}(\emptyset)}{\sum_{i=1}^{s} \varrho_{M_i}(\emptyset)} = 1 - c_{k_1}. \tag{2.45}$$

By (2.3), c_{k_2} can be written as

$$c_{k_2} = 1 - \min_{J \in \hat{X}} \left\{ \frac{\varrho_J(X \setminus J)}{\varrho_J(\emptyset)} \right\}. \tag{2.46}$$

By (2.45) and (2.46), we have $c_{k_2} \leq 1 - (1 - c_{k_1}) = c_{k_1}$. □

Remark 2.2.12. *When $k_1 = 1$ and $k_2 = k$, Theorem 2.2.8 reduces to Theorem 2.2.7. However, the proof of Theorem 2.2.7 can be used only to prove the case when k_1 divides k_2 in Theorem 2.2.8. This is why we have chosen to separate the two theorems.*

2.3 Examples

In this section, we consider a task scheduling problem and an adaptive sensing problem to illustrate our results. Specially, we demonstrate that the total curvature c_k decreases in k and the performance bound for a uniform matroid increases in k under the condition that k divides K.

2.3.1 Task Scheduling

As a canonical example for problem (2.2), we consider the task scheduling problem posed in [3], which was also analyzed in [4] and [57]. In this problem, there are n subtasks and a set X of N agents. At each stage, a subtask i is assigned to an agent a, who accomplishes the task with probability $p_i(a)$. Let $X_i(\{a_1, a_2, \ldots, a_k\})$ denote the Bernoulli random variable that signifies whether or not subtask i has been accomplished after performing the set of agents $\{a_1, a_2, \ldots, a_k\}$ over k stages. Then $\frac{1}{n}\sum_{i=1}^{n} X_i(\{a_1, a_2, \ldots, a_k\})$ is the fraction of subtasks accomplished after k stages by employing agents $\{a_1, a_2, \ldots, a_k\}$. The objective function f for this problem is the expected value of this fraction, which can be written as

$$f(\{a_1, \ldots, a_k\}) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \prod_{j=1}^{k}(1 - p_i(a_j))\right).$$

Assume that $p_i(a) > 0$ for any $a \in X$. Then it is easy to check that f is nondecreasing. Therefore, when $\mathcal{I} = \{S \subseteq X : |S| \leq K\}$, this problem has an optimal solution of length K. Also, it is easy to check that f has the diminishing-return property and $f(\emptyset) = 0$. Thus, f is a polymatroid set function.

For convenience, we only consider the special case $n = 1$; our analysis can be generalized to any $n \geq 2$. For $n = 1$, we have

$$f(\{a_1, \ldots, a_k\}) = 1 - \prod_{j=1}^{k} (1 - p(a_j)),$$

where $p(\cdot) = p_1(\cdot)$.

Let us order the elements of X as $a_{[1]}, a_{[2]}, \ldots, a_{[N]}$ such that

$$0 < p(a_{[1]}) \leq p(a_{[2]}) \leq \ldots \leq p(a_{[N]}) \leq 1.$$

Then by the definition of the total curvature c_k, we have

$$c_k = \max_{i_1,\ldots,i_k \in X} \left\{ 1 - \frac{f(X) - f(X \setminus \{i_1, \ldots, i_k\})}{f(\{i_1, \ldots, i_k\}) - f(\emptyset)} \right\} = 1 - \prod_{l=k+1}^{N} (1 - p(a_{[l]})).$$

To numerically evaluate the relevant quantities here, we randomly generate a set of $\{p(a_i)\}_{i=1}^{30}$. In Figure 2.2, we consider $K = 20$, and batch sizes $k = 1, 2, \ldots, 10$. From the expression of c_k, we can see that c_k is nonincreasing in k, but when N is large, c_k is close to 1 for each k. Figure 2.2 shows that the exponential bound for $k = 3, 6, 8, 9$ is worse than that for $k = 1, 2$, which illustrates our earlier remark that the exponential bound for the uniform matroid case is not necessarily e in k even though c_k is monotone in k. Figure 2.2 also shows that the exponential bound $\frac{1}{c_k}(1 - (1 - \frac{c_k}{l+1}\frac{m}{k})(1 - \frac{c_k}{l+1})^l$ coincides with $\frac{1}{c_k}(1 - (1 - \frac{c_k}{l+1})^{l+1})$ for $k = 1, 2, 4, 5, 10$ and it is nondecreasing in k, which illustrates our remark that the exponential bound is nondecreasing in k under the condition that k divides K.

Owing to the nature of the total curvature for this example, it is not easy to see that c_k is nonincreasing in k (all c_k values here are very close to 1). The next example will illustrate that the total curvature does decrease in k and again demonstrate our claim that the exponential bound

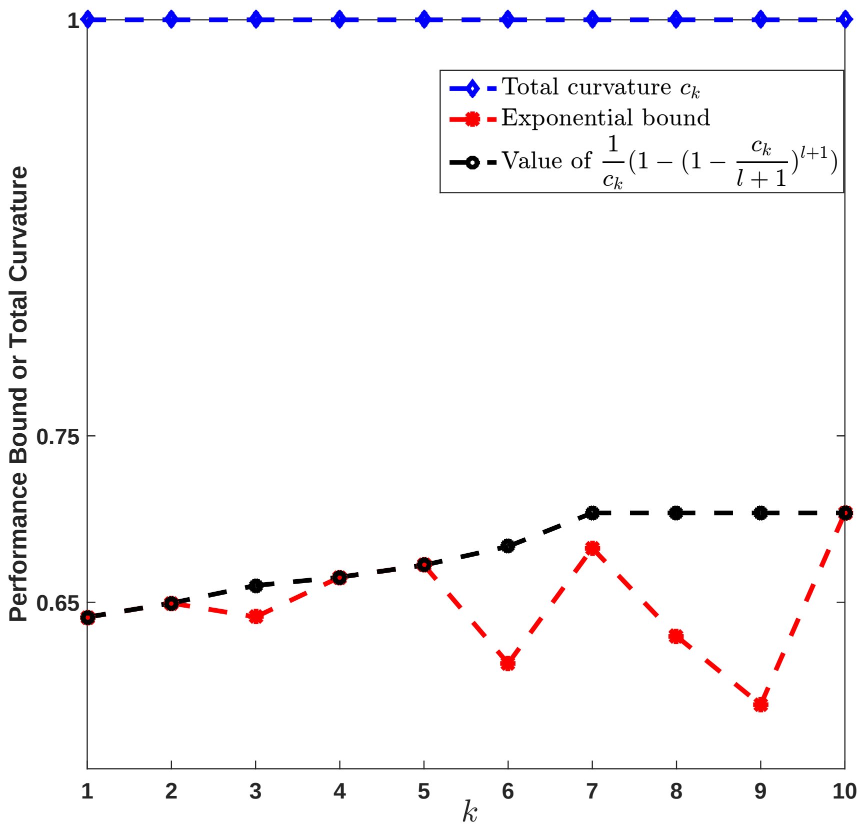

Figure 2.2: Task scheduling example

for the uniform matroid case is not necessarily monotone in k but it is monotone in k under the condition that k divides K.

2.3.2 Adaptive Sensing

As our second example application, we consider the adaptive sensing design problem posed in [30] and [4]. Consider a signal of interest $x \in \mathbb{R}^2$ with normal prior distribution $\mathcal{N}(0, I)$, where I is the 2×2 identity matrix; our analysis easily generalizes to dimensions larger than 2. Let $\mathbb{B} = \{\text{Diag}(\sqrt{b}, \sqrt{1-b}) : b \in \{b_1, \ldots, b_N\}\}$, where $b_i \in [0.5, 1]$ for $1 \leq i \leq N$. At each stage i, we make a measurement y_i of the form

$$y_i = B_i x + w_i,$$

where $B_i \in \mathbb{B}$ and w_i represents i.i.d. Gaussian measurement noise with mean zero and covariance $\sigma^2 I$, independent of x.

The objective function f for this problem is the information gain, which can be written as

$$f(\{B_1, \ldots, B_k\}) = H_0 - H_k.$$

Here, $H_0 = \frac{N}{2}\log(2\pi e)$ is the entropy of the prior distribution of x and H_k is the entropy of the posterior distribution of x given $\{y_i\}_{i=1}^k$; that is,

$$H_k = \frac{1}{2}\log\det(P_k) + \frac{N}{2}\log(2\pi e),$$

where

$$P_k = \left(P_{k-1}^{-1} + \frac{1}{\sigma^2} B_k^T B_k\right)^{-1}$$

is the posterior covariance of x given $\{y_i\}_{i=1}^k$ [30].

The objective is to choose a set of measurement matrices $\{B_i^*\}_{i=1}^K$, $B_i^* \in \mathbb{B}$, to maximize the information gain $f(\{B_1, \ldots, B_K\}) = H_0 - H_K$. It is easy to check that f is nondecreasing, submodular, and $f(\emptyset) = 0$; i.e., f is a polymatroid set function.

For convenience, let $\sigma = 1$. Then, we have

$$\begin{aligned} c_k &= \max_{J_k \subseteq X, |J_k| = k} \left\{1 - \frac{f(X) - f(X \setminus J_k)}{f(J_k)}\right\} \\ &= \max_{J_k \subseteq X, |J_k| = k} \left\{1 - \frac{\log(st) - \log\left(s - \sum_{i: e_i \in J_k} e_i\right)\left(t - \sum_{i: e_i \in J_k} (1 - e_i)\right)}{\log\left(1 + \sum_{i: e_i \in J_k} e_i\right)\left(1 + \sum_{i: e_i \in J_k} (1 - e_i)\right)}\right\}, \end{aligned}$$

where $X = \{B_1, \ldots, B_N\}$, $s = 1 + \sum_{i=1}^N e_i$, and $t = 1 + \sum_{i=1}^N (1 - e_i)$.

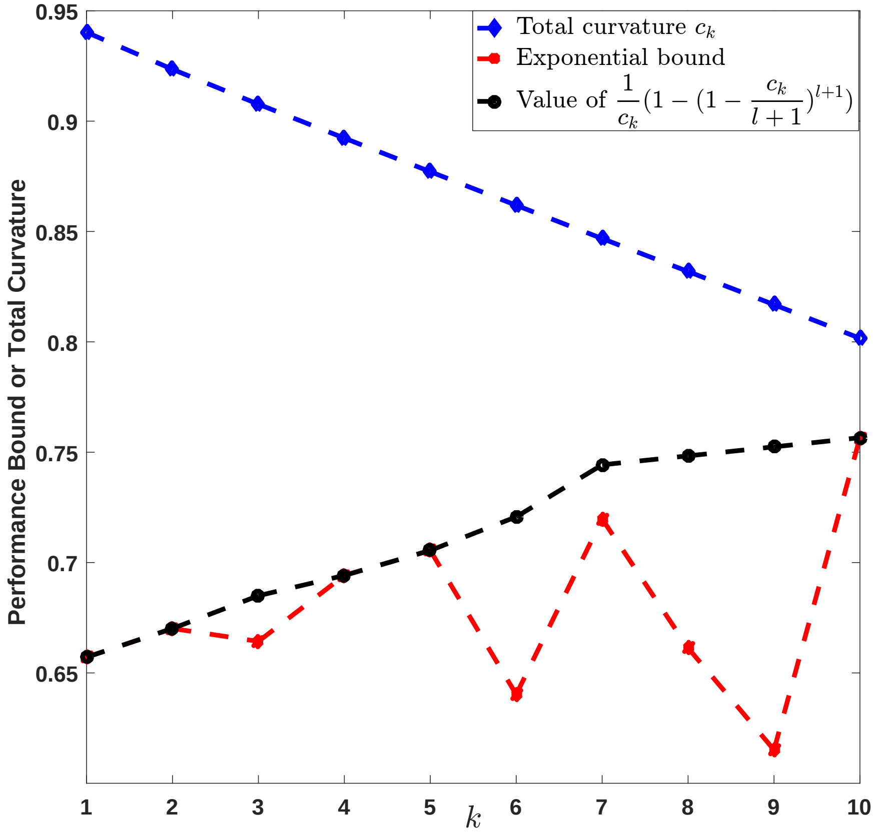

Figure 2.3: Adaptive sensing example

To numerically evaluate the relevant quantities here, we randomly generate a set of $\{e_i\}_{i=1}^{30}$. We first still consider $K = 20$ for $k = 1, \ldots, 10$ in Figure 2.3. Figure 2.3 shows that the total curvature decreases in k, while the exponential bound for the uniform matroid case only increases for $k = 1, 2, 4, 5, 7, 10$ and the bound for $k = 3, 6, 8, 9$ is worse than that for $k = 1, 2$. This illustrates that the exponential bound for the uniform matroid case is not necessarily monotone in k.

Next, we consider $K = 24$ for $k = 1, 2, 3, 4, 6, 8$ in Figure 2.4. Figure 2.4 shows that the curvature decreases in k and the exponential bound increases in k since k divides K for $k = 1, 2, 3, 4, 6, 8$, which again demonstrates our claim that c_k decreases in k and the exponential bound increases in k under the condition that k divides K.

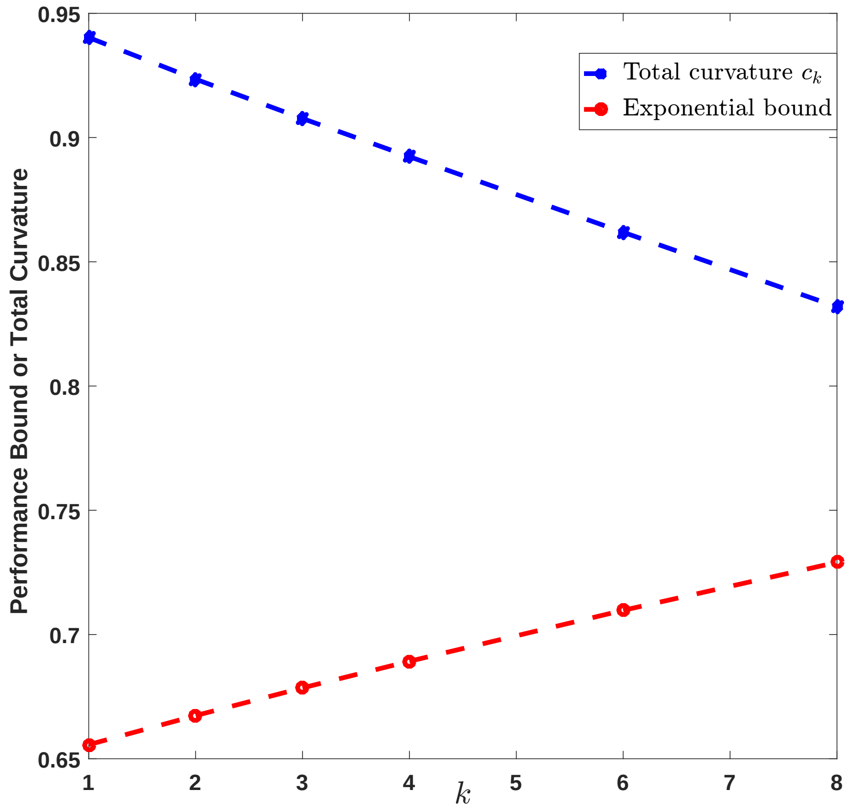

Figure 2.4: Adaptive sensing example

2.4 Discussion on Matroid Preservation

Suppose that $(X, \mathcal{I})$ is a uniform matroid. In this appendix, we will provide an example to prove that the collection of subsets of X of size k satisfying the constraint $\mathcal{I}$ (i.e., actions in the k-batch greedy strategy) is not in general a matroid. This shows that lifting does not work; i.e., it is not in general possible to appeal to bounds for the 1-batch greedy strategy to derive bounds for the k-batch greedy strategy. For convenience, we assume that k divides the uniform matroid rank K.

Recall that for a matroid $(X, \mathcal{I})$, we have the following two properties:

i. For all $B \in \mathcal{I}$, any set $A \subseteq B$ is also in $\mathcal{I}$.

ii. For any $A, B \in \mathcal{I}$, if $|B| > |A|$, then there exists $j \in B \setminus A$ such that $A \cup \{j\} \in \mathcal{I}$.

To apply lifting, first fix k. We will define a pair $(Y, \mathcal{J})$ such that Y is the "ground set" of all k-element subsets of X: $Y = \{y : y = \{a_1, \ldots, a_k\}, k$ is given, and $a_i \in X\}$. Next, $\mathcal{J}$ is the set of all subsets of Y such that their elements are disjoint and the union of their elements lies in $\mathcal{I}$. The following example shows that $(Y, \mathcal{J})$ constructed this way is not in general a matroid.

Example 2.4.1. *Fix $k = 2$. Let $X = \{a, b, c, d\}$, and $\mathcal{I}$ be the power set of X (a special case of a uniform matroid, with rank $K = 4$). We have $Y = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}\}$. Let $\mathcal{J}$ be as defined above.*

We will now prove that $(Y, \mathcal{J})$ does not satisfy property ii above. To see this, consider $A = \{\{a, b\}\} \in \mathcal{J}$ and $B = \{\{a, c\}, \{b, d\}\} \in \mathcal{J}$. We have $|A| = 1$ and $|B| = 2$. Notice that $\{a, b\} \cap \{a, c\} \neq \emptyset$ and $\{a, b\} \cap \{b, d\} \neq \emptyset$. So, in this case clearly there does not exist $j \in B \setminus A$ such that $A \cup \{j\} \in \mathcal{J}$. Hence, property ii fails and $(Y, \mathcal{J})$ is not a matroid.

2.5 Comparing Different k-Batch Greedy Strategies

It is tempting to think that the k-batch ($k \geq 2$) greedy strategy always outperforms the 1-batch greedy strategy. In fact, this is false. To show this, we will provide two examples based on the maximum K-coverage problem, which was considered in [51] to demonstrate via Monte Carlo simulations that the 1-batch greedy strategy can perform better than the 2-batch greedy strategy. The maximum K-coverage problem is to select at most K sets from a collection of sets such that the union of the selected sets has the maximum number of elements. Example 2.5.1 below is to choose at most 3 sets from a collection of 5 sets, and Example 2.5.2 is to choose at most 4 sets from a collection of 6 sets. In contrast to [51], our examples are not based on simulation, but are analytical counterexamples.

Example 2.5.1. *Fix $K = 3$ and let the sets to be selected be $S_1 = \{a, f\}$, $S_2 = \{f\}$, $S_3 = \{a, b, g\}$, $S_4 = \{c, f, g\}$, and $S_5 = \{e, g, h\}$.*

For the 1-batch greedy strategy, one solution is $\{S_3, S_4, S_5\}$, and the union of the selected sets is $S_3 \cup S_4 \cup S_5 = \{a, b, c, e, f, g, h\}$. For the 2-batch greedy strategy, one solution is $\{S_1, S_5, S_3\}$, and the union of the selected sets is $S_1 \cup S_5 \cup S_3 = \{a, b, e, f, g, h\}$. It is easy to see that $|S_3 \cup S_4 \cup S_5| = 7 > |S_1 \cup S_5 \cup S_3| = 6$.

Example 2.5.2. *Fix $K = 4$ and let the sets to be selected be $S_1 = \{h, i, j\}, S_2 = \{b, e, i, j\}$, $S_3 = \{c, d, e, h\}, S_4 = \{b, d, f, h, i\}, S_5 = \{a, h, i, j\}$, and $S_6 = \{c, g, i\}$.*

For the 1-batch greedy strategy, one solution is $\{S_4, S_2, S_6, S_5\}$, and the union of the selected sets is $S_4 \cup S_2 \cup S_6 \cup S_5 = \{a, b, c, d, e, f, g, h, i, j\}$. For the 2-batch greedy strategy, one solution is $\{S_2 \cup S_3 \cup S_4 \cup S_5\}$, and the union of the selected sets is $S_2 \cup S_3 \cup S_4 \cup S_5 = \{a, b, c, d, e, f, h, i, j\}$. It is easy to see that $|S_4 \cup S_2 \cup S_6 \cup S_5| = 10 > |S_2 \cup S_3 \cup S_4 \cup S_5| = 9$.

For the two examples above, it is easy to check that their 1-batch and 2-batch greedy solutions are not unique. For Example 2.5.1, the 1-batch greedy solution $\{S_3, S_4, S_5\}$ is also one solution of the 2-batch greedy strategy. If we choose $\{S_3, S_4, S_5\}$ instead of $\{S_1, S_5, S_3\}$ as the solution of the 2-batch greedy strategy, then the 1-batch greedy strategy has the same performance as the 2-batch greedy strategy in this case. For Example 2.5.2, the 1-batch greedy solution $\{S_4, S_2, S_6, S_5\}$ is also one solution of the 2-batch greedy strategy. So we can say that for the k-batch greedy strategy, its solution is not unique. However, our harmonic bound under general matroid constraints and exponential bound under uniform matroid constraints are both universal, which means that the harmonic bound holds for any k-batch greedy solution under general matroid constraints and the exponential bound holds for any k-batch greedy solution under uniform matroid constraints.

Chapter 3

Improved Bounds for Greedy Strategy in Set Optimization

In this chapter, we still consider problem (2.2) in Chapter 2. For convenience of reference, we rewrite the problem which is to find a set in $\mathcal{I}$ to maximize the objective function f as follows:

$$\begin{aligned} &\text{maximize} \;\; f(M) \\ &\text{subject to} \;\; M \in \mathcal{I}. \end{aligned} \tag{3.1}$$

Suppose that the objective function f in problem (3.1) is a polymatroid function and the rank of the matroid $(X, \mathcal{I})$ is K. By the augmentation property of a matroid and the monotoneity of f, any optimal solution can be extended to a set of size K. By the definition of the greedy strategy (see Section 2.1.3), any greedy solution is of size K. For the greedy strategy, under a general matroid constraint and a uniform matroid constraint, the performance bounds $1/(1+c)$ and $(1-(1-c/K)^K)/c$ from [34] are the best so far, respectively, in terms of the total curvature c. However, the total curvature c, by definition, depends on the function values on sets outside the matroid $(X, \mathcal{I})$. This gives rise to two possible issues when applying existing bounding results involving the total curvature c:

1. If we are given a function f defined only on $\mathcal{I}$, then problem (1) still makes sense, but the total curvature is no longer well defined. This means that the existing results involving the total curvature do not apply. But this surely is puzzling: if the optimization problem (1) is perfectly well defined, why should the bounds no longer apply?

2. Even if the function f is defined on the entire 2^X, the fact that the total curvature c involves sets outside the matroid is puzzling. Specifically, if the optimization problem (1) involves

only sets in the matroid, why should the bounding results rely on a quantity c that depends on sets outside the matroid?

The two reasons above motivate us to investigate more applicable bounds involving only sets in the matroid.

In Section 3.1, we first introduce definitions of polymatroid functions, matroids, and curvature, and then we review performance bounds in terms of the total curvature from [34]. In Section 3.2.1, we prove that any monotone set function defined on the matroid can be extended to one defined on the entire power set and the extended function can be expressed in a certain form. In Section 3.2.2, we provide necessary and sufficient conditions for the existence of an incremental extension of a polymatroid function defined on the uniform matroid of rank k to one defined on the uniform matroid of rank $k+1$. In Section 3.2.3, we introduce a particular extension we call the majorizing extension and explore what kinds of polymatroid functions can be majorizingly extended to ones defined on the whole power set. In Section 3.2.4, we provide an algorithm for constructing the extension of a polymatroid function defined on a matroid to the entire power set. In Section 3.3, we define the partial curvature involving only sets in the matroid and obtain improved bounds in terms of the partial curvature subject to certain necessary and sufficient conditions. In Section 3.4, we illustrate our results by considering a task scheduling problem and an adaptive sensing problem.

The results in this chapter were published in [58, 59].

3.1 Preliminaries

3.1.1 Polymatroid Functions and Curvature

The definitions and terminology in this section were introduced in Chapter 2, but are reviewed again here for convenience of reference. Let X be a finite ground set of actions, and $\mathcal{I}$ be a non-empty collection of subsets of X. Given a pair $(X, \mathcal{I})$, the collection $\mathcal{I}$ is said to be hereditary if it satisfies property i below and has the augmentation property if it satisfies property ii below:

i. (Hereditary) For all $B \in \mathcal{I}$, any set $A \subseteq B$ is also in $\mathcal{I}$.

ii. (Augmentation) For any $A, B \in \mathcal{I}$, if $|B| > |A|$, then there exists $j \in B \setminus A$ such that $A \cup \{j\} \in \mathcal{I}$.

The pair $(X, \mathcal{I})$ is called a matroid if it satisfies both properties i and ii. The pair $(X, \mathcal{I})$ is called a uniform matroid when $\mathcal{I} = \{S \subseteq X : |S| \leq K\}$ for a given K, called the rank of $(X, \mathcal{I})$. In general, the rank of a matroid $(X, \mathcal{I})$ is the cardinality of its maximal set.

Let 2^X denote the power set of X, and define a set function f: $2^X \to \mathbb{R}$. The set function f is said to be monotone and submodular if it satisfies properties 1 and 2 below, respectively:

1. (Monotone) For any $A \subseteq B \subseteq X$, $f(A) \leq f(B)$.

2. (Submodular) For any $A \subseteq B \subseteq X$ and $j \in X \setminus B$, $f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B)$.

A set function f: $2^X \to \mathbb{R}$ is called a polymatroid function [55] if it is monotone, submodular, and $f(\emptyset) = 0$, where $\emptyset$ denotes the empty set. The submodularity in property 2 means that the additional value accruing from an extra action decreases as the size of the input set increases. This property is also called the diminishing-return property in economics.

The total curvature [34] of a set function f is defined as

$$c(f) = \max_{\substack{j \in X \\ f(\{j\}) \neq f(\emptyset)}} \left\{ 1 - \frac{f(X) - f(X \setminus \{j\})}{f(\{j\}) - f(\emptyset)} \right\}. \tag{3.2}$$

For convenience, we use c to denote $c(f)$ when there is no ambiguity. Note that $0 \leq c \leq 1$ when f is a polymatroid function, and $c = 0$ if and only if f is additive, i.e., for any set $A \subseteq X$, $f(A) = \sum_{i \in A} f(\{i\})$. When $c = 0$, it is easy to check that the greedy strategy coincides with the optimal strategy. So in the rest of the paper, when we assume that f is a polymatroid function, we only consider $c \in (0, 1]$.

3.1.2 Performance Bounds in Terms of Total Curvature

In this section, we review two theorems from [34], which bound the performance of the greedy strategy using the total curvature c for general matroid constraints and uniform matroid constraints. We will use these two theorems to derive bounds in Section 3.3.

We first define optimal and greedy solutions for (3.1) as follows:

Optimal solution: A set O is called an optimal solution of (3.1) if

$$O \in \underset{M \in \mathcal{I}}{\operatorname{argmax}} f(M),$$

where the right-hand side denotes the collection of arguments that maximize $f(\cdot)$ on $\mathcal{I}$. Note that there may exist more than one optimal solution for problem (3.1). When $(X, \mathcal{I})$ is a matroid of rank K, then any optimal solution can be extended to a set of size K because of the augmentation property of the matroid and the monotoneity of the set function f.

Greedy solution: A set $G = \{g_1, g_2, \dots, g_K\}$ is called a greedy solution of (3.1) if

$$g_1 \in \underset{\{g\} \in \mathcal{I}}{\operatorname{argmax}} f(\{g\}),$$

and for $i = 2, \dots, K$,

$$g_i \in \underset{\substack{g \in X \\ \{g_1, \dots, g_{i-1}, g\} \in \mathcal{I}}}{\operatorname{argmax}} f(\{g_1, g_2, \dots, g_{i-1}, g\}).$$

Note that there may exist more than one greedy solution for problem (3.1).

Theorem 3.1.1. *[34] Let $(X, \mathcal{I})$ be a matroid and $f\colon 2^X \to \mathbb{R}$ be a polymatroid function with total curvature c. Then, any greedy solution G satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{1+c}.$$

When f is a polymatroid function, we have $c \in (0, 1]$, and therefore $1/(1 + c) \in [1/2, 1)$. Theorem 3.1.1 applies to any matroid. This means that the bound $1/(1 + c)$ holds for a uniform matroid too. Theorem 3.1.2 below provides a tighter bound when $(X, \mathcal{I})$ is a uniform matroid.

Theorem 3.1.2. *[34] Let $(X, \mathcal{I})$ be a uniform matroid of rank K and $f\colon 2^X \to \mathbb{R}$ be a polymatroid function with total curvature c. Then, any greedy solution G satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{c}\left(1-\left(1-\frac{c}{K}\right)^K\right) > \frac{1}{c}\left(1-e^{-c}\right).$$

The function $(1-(1-c/K)^K)/c$ is nonincreasing in K for $c \in (0,1]$ and $(1-(1-c/K)^K)/c \searrow (1-e^{-c})/c$ when $K \to \infty$; therefore, $(1-(1-c/K)^K)/c > (1-e^{-c})/c$ when f is a polymatroid function. Also it is easy to check that $(1-e^{-c})/c > 1/(1+c)$ for $c \in (0,1]$, which implies that the bound $(1-(1-c/K)^K)/c$ is stronger than the bound $1/(1+c)$ in Theorem 3.1.1.

The bounds in Theorems 3.1.1 and 3.1.2 involve sets not in the matroid, so as stated they do not apply to optimization problems whose objective function is only defined for sets in the matroid. In the following section, we will explore the extension of polymatroid functions that yield to the bounds in Theorems 3.1.1 and 3.1.2.

3.2 Function Extension

3.2.1 Monotone Extension

The following proposition states that any monotone set function defined on the matroid $(X, \mathcal{I})$ can be extended to one defined on the entire power set 2^X, and the extended function can be expressed in a certain form.

Proposition 3.2.1. *Let $(X, \mathcal{I})$ be a matroid of rank K and $f : \mathcal{I} \to \mathbb{R}$ be a monotone set function. Then there exists a monotone set function $g : 2^X \to \mathbb{R}$ satisfying the following conditions:*

a. $g(A) = f(A)$ for all $A \in \mathcal{I}$.

b. g is monotone on 2^X.

Moreover, any function $g : 2^X \to \mathbb{R}$ satisfying the above two conditions can be expressed as

$$g(A) = \begin{cases} f(A), & A \in \mathcal{I}, \\ g(B^*) + d_A, & A \notin \mathcal{I}, \end{cases} \tag{3.3}$$

where

$$B^* \in \operatorname*{argmax}_{\substack{B: B \subset A \\ |B| = |A|-1}} g(B) \tag{3.4}$$

and d_A is a nonnegative number.

Proof. Condition a can be satisfied by construction: first set

$$g(A) = f(A) \tag{3.5}$$

for all $A \in \mathcal{I}$. To prove that there exists a monotone set function g defined on the entire power set 2^X satisfying both conditions a and b, we prove the following statement by induction: There exists a set function g of the form

$$g(A) = \begin{cases} f(A), & A \in \mathcal{I}, \\ g(B^*) + d_A, & A \notin \mathcal{I}, \end{cases} \tag{3.6}$$

such that g is monotone for sets of size up to l ($l \leq K$), where B^* is given in (3.4) and d_A is a nonnegative number.

First, we prove that the above statement holds for $l = 1$. For g to be monotone for sets of size up to 1, it suffices to have that $g(A) \geq 0$ for any set $A \in 2^X$ with $|A| = 1$. For $A \in \mathcal{I}$, by (3.5) we have that $g(A) = f(A) \geq 0$. For $A \notin \mathcal{I}$, it suffices to set $g(A) = d_A$, where d_A is any nonnegaative number. Therefore, the above statement holds for $l = 1$. Assume that the above statement holds for $l = k$. We prove that it also holds for $l = k + 1$. For this, it suffices to prove that for any $A \in 2^X$ with $|A| = k + 1$ and any $B \subset A$, we have that $g(A) \geq g(B)$.

Consider any set $A \in \mathcal{I}$ with $|A| = k + 1$. By (3.5) we have that $g(A) = f(A)$. For any set $B \subset A$, by the hereditary property of a matroid, we have that $B \in \mathcal{I}$, which implies that $g(B) = f(B)$. So for any set $A \in \mathcal{I}$ with $|A| = k + 1$ and any set $B \subset A$, by the condition that f is monotone on $\mathcal{I}$, we have that $g(A) \geq g(B)$.

Consider any set $A \notin \mathcal{I}$ with $|A| = k + 1$. By the induction hypothesis for $l = k$, we have that for any set $B \subset A$ with $|B| = k$, $g(B)$ is well defined. Set $d_A \geq 0$ and

$$B^* \in \operatorname*{argmax}_{\substack{B:B\subset A\\|B|=|A|-1}} g(B),$$

and then define

$$g(A) = g(B^*) + d_A.$$

We have that

$$g(A) \geq g(B) \tag{3.7}$$

for any set $B \subset A$ with $|B| = k$. For any set $B \subset A$ with $|B| < k$, there must exist a set A_k with $|A_k| = k$ such that $B \subset A_k \subset A$. By the induction hypothesis $l = k$ and (3.7), we have that

$$g(A) \geq g(A_k) \geq g(B). \tag{3.8}$$

Combining (3.7) and (3.8), for any set $A \notin \mathcal{I}$ with $|A| = k + 1$ and any set $B \subset A$, we have that $g(A) \geq g(B)$. Therefore, (3.6)) holds for $l = k + 1$.

We have so far shown that there exists a monotone set function $g : 2^X \to \mathbb{R}$ satisfying conditions a and b. Next we prove that any monotone set function $g : 2^X \to \mathbb{R}$ satisfying conditions a and b can be expressed as in (3.3).

If g satisfies condition a, then we have that

$$g(A) = f(A),\ \forall A \in \mathcal{I}. \tag{3.9}$$

If g satisfies condition b, then for any set $A \notin \mathcal{I}$, we have that

$$g(A) \geq g(B^*),$$

which implies that there exists a nonnegative number d_A such that

$$g(A) = g(B^*) + d_A,\ \forall A \notin \mathcal{I}. \tag{3.10}$$

Combining (3.9) and (3.10), we have that any monotone set function $g : 2^X \to \mathbb{R}$ satisfying conditions a and b can be expressed by (3.3). $\square$

In Proposition 3.2.1, when $A \notin \mathcal{I}$, we define $g(A)$ using B^* as defined in (3.4). But we are not restricted to using B^* as the following lemma shows.

Lemma 3.2.2. *Assume that g is a monotone set function defined on the uniform matroid of rank k. Then, for any set A with $|A| = k+1$, there exist nonnegative numbers $d_1, d_2, \ldots, d_M$ such that*

$$g(A) = g(A_1) + d_1 = g(A_2) + d_2 = \cdots = g(A_M) + d_M,$$

where $M = 2^{k+1} - 2$ and $A_1, A_2, \ldots, A_M$ denote all nonempty strict subsets of A.

Proof. Without loss of generality, let

$$A_M \in \operatorname*{argmax}_{B : B \subset A, |B| = |A|-1} g(B).$$

By Proposition 3.2.1, we have that there exist $d_M \geq 0$ such that $g(A) = g(A_M) + d_M$. Then, for any $i = 1, \ldots, M-1$, setting $d_i = d_M + g(A_M) - g(A_i)$ results in

$$g(A) = g(A_1) + d_1 = g(A_2) + d_2 = \cdots = g(A_M) + d_M,$$

where $d_i \geq 0$, because $d_M \geq 0$ and $g(A_M) \geq g(A_i)$ for $i = 1, \cdots, M-1$. $\square$

3.2.2 Polymatroid Extension: From Uniform Matroid to Power Set

We now turn our attension to extending polymatroid functions. The authors of [60] pointed out that there are cases where a polymatroid function defined on a matroid cannot be extended to one that is defined on the entire power set. In the theorem below, we give necessary and sufficient conditions for the existence of an extension of a polymatroid function defined on the uniform matroid of rank k to the uniform matroid of rank $k+1$.

Theorem 3.2.3. *Let $f : \mathcal{I} \to \mathbb{R}$ be a polymatroid function defined on the uniform matroid of rank k. Then f can be extended to a polymatroid function g defined on the uniform matroid of rank $k+1$ if and only if for any $A \subseteq X$ with $|A| = k+1$, any $B \subset A$ with $|B| = k$, and any $a \in B$,*

$$f(B) - f(B \setminus \{a\}) \geq f(B^*) - f(A \setminus \{a\}), \tag{3.11}$$

where

$$B^* \in \operatorname*{argmax}_{B: B \subset A, |B| = k} f(B). \tag{3.12}$$

Proof. $\rightarrow$

In this direction, we need to prove that (3.11) holds if g is an extended polymatroid function defined on the uniform matroid of rank $k+1$. If g is a polymatroid function, we have that g is monotone and submodular. If g is monotone, then for any set $A \notin \mathcal{I}$ with $|A| = k+1$, we have

$$g(A) \geq g(B^*). \tag{3.13}$$

If g is submodular, then for any set A , any set $B \subset A$ with $|B| = k$, and any action $a \in B$, we have

$$g(B) - g(B \setminus \{a\}) \geq g(A) - g(A \setminus \{a\}). \tag{3.14}$$

Combining (3.13) and (3.14), we have

$$g(B) - g(B \setminus \{a\}) \geq g(B^*) - g(A \setminus \{a\}).$$

Because g is an extended function of f, we have that $g(B) = f(B)$, $g(B^*) = f(B^*)$, $g(B \setminus \{a\}) = f(B \setminus \{a\})$, and $g(A \setminus \{a\}) = f(A \setminus \{a\})$. Then the above inequality becomes

$$f(B) - f(B \setminus \{a\}) \geq f(B^*) - f(A \setminus \{a\}).$$

which means that (3.11) holds.

$\leftarrow$

In this direction, we prove that if (3.11) holds, then there exists a polymatroid function g defined on the uniform matroid of rank $k+1$ that agrees with f on the uniform matroid of rank k.

By Proposition 3.2.1, we have that there exists an extended monotone set function g of the following form defined on the uniform matroid of rank $k+1$:

$$g(A) = \begin{cases} f(A), & |A| \leq k, \\ f(B^*) + d_A, & |A| = k+1, \end{cases} \tag{3.15}$$

where B^* is defined as in (3.12) and d_A is nonnegative.

We will prove that there exists d_A for any $A \subset X$ with $|A| = k+1$ such that g defined in (3.15) satisfies $g(\emptyset) = 0$ and g is submodular on 2^X.

Because f is a polymatroid function on the uniform matroid of k and $g(A) = f(A)$ for any $A \subseteq X$ with $|A| \leq k$, we have that $g(\emptyset) = f(\emptyset) = 0$. For g to be submodular on the uniform matroid of rank $k+1$, it suffices to have that for any $A \subseteq X$ with $|A| = k+1$, any $B \subset A$ with $|B| = k$, and any $a \in B$

$$g(B) - g(B \setminus \{a\}) \geq g(A) - g(A \setminus \{a\}). \tag{3.16}$$

For any $A \subseteq X$ with $|A| = k+1$, by (3.15), we have that $g(A) = f(B^*) + d_A$, where $d_A \geq 0$. The inequality (3.11) implies that $f(B) - f(B^*) + f(A \setminus \{a\}) - f(B \setminus \{a\}) \geq 0$. So only if we set d_A to satisfy

$$0 \leq d_A \leq \min_{B: B \subset A, |B|=k \text{ and } a: a \in B} \{f(B) - f(B^*) + f(A \setminus \{a\}) - f(B \setminus \{a\})\}, \tag{3.17}$$

we have that $g(A) \leq f(B) - f(B \setminus \{a\}) + f(A \setminus \{a\})$, which implies that (3.16) holds.

This completes the proof. $\square$

Remark 3.2.1. *Theorem 3.2.3 provides necessary and sufficient conditions for the existence of an extension of a polymatroid function defined on the uniform matroid of rank k to the uniform matroid of rank $k+1$. We will show that the function in the following example, taken from [60], does not have an extension because (3.11) is not satisfied.*

Example 1: Let $X = \{1, 2, 3\}$ and $\mathcal{I} = \{A : A \in X$ and $|A| \leq 2\}$. Define $f : \mathcal{I} \to \mathbb{R}$ as follows:

$f(\emptyset) = 0,$

$f(\{1\}) = f(\{2\}) = f(\{3\}) = 1,$

$f(\{1, 2\}) = f(\{1, 3\}) = 1,$ *and* $f(\{2, 3\}) = 2.$

It is easy to show that the above function f is a polymatroid function on the uniform matroid of rank 2. But as we now show, f cannot be extended to a polymatroid function g on the uniform matroid of rank 3 which is also the power set.

Setting $A = X$, by (3.15), it is easy to see that $B^* = \{2, 3\}$. Then we have $g(X) = f(\{2, 3\}) + d_X$, where $d_X \geq 0$. If (3.11) holds for $A = X$, $B = \{1, 2\}$, and $\{a\} = \{2\}$, we have the following inequality:

$$f(\{1, 2\}) - f(\{2, 3\}) + f(\{1, 3\}) - f(\{1\}) \geq 0.$$

However,

$$f(\{1, 2\}) - f(\{2, 3\}) + f(\{1, 3\}) - f(\{1\}) = -1 < 0.$$

We conclude that (3.11) does not hold always. Then by Theorem 3.2.3, we have that the polymatroid function f defined above does not have an extended polymatroid function defined on the whole power set.

3.2.3 Majorizing Extension

Theorem 3.2.3 and Proposition 3.2.1 together provide us an algorithm to extend a polymatroid function f defined on the uniform matroid of rank k to a polymatroid function g defined on the uniform matroid of rank $k+1$. The procedure is to construct g as in (3.15) with d_A satisfying

(3.17). By (3.17), if for any A with $|A| = k + 1$,

$$\min_{B:B\subset A,|B|=k \text{ and } a:a\in B}\{f(B) - f(B^*) + f(A \setminus \{a\}) - f(B \setminus \{a\})\} \geq 0,$$

then f can be extended to g. We say that f is chap3majorizingly extended to g if for any A with $|A| = k + 1$, we set

$$d_A = \min_{B:B\subset A,|B|=k \text{ and } a:a\in B}\{f(B) - f(B^*) + f(A \setminus \{a\}) - f(B \setminus \{a\})\}. \tag{3.18}$$

Remark 3.2.2. *The reason we are calling this particular construction of g a majorizing extension is that the sequence $\{d_A\}$ (indexed by A) majorizes any other sequence $\{d'_A\}$ whose elements satisfy (3.17), because $d_A \geq d'_A$ for any $A \subseteq X$.*

We just introduced the definition of a majorizing extension. We wish to explore what kind of polymatroid functions can be majorizingly extended to ones defined on the whole power set. The following theorem states that a polymatroid function defined on the uniform matroid of rank 1 can be majorizingly extended to one defined on the power set, and the extended function is additive.

Theorem 3.2.4. *Let X be a ground set and f a polymatroid function defined on the uniform matroid of rank 1. Then f can be majorizingly extended to a polymatroid function g defined on the power set 2^X with*

$$g(\{x_1, x_2, \ldots, x_k\}) = \sum_{j=1}^{k} f(\{x_j\})$$

for any set $\{x_1, x_2, \ldots, x_k\} \subseteq X$.

Proof. We will prove the theorem by induction on k. Without loss of generality, we assume for convenience that $X = \{1, 2, \ldots, N\}$ and $f(\{1\}) \leq f(\{2\}) \leq \cdots \leq f(\{N\})$.

First, we prove the claim for $k = 2$, i.e., $g(\{x_1, x_2\}) = f(\{x_1\}) + f(\{x_2\})$ for any $\{x_1, x_2\} \subseteq X$ $(x_1 < x_2)$. By the assumption above and (3.15), we have that

$$g(\{x_1, x_2\}) = f(\{x_2\}) + d_{\{x_1,x_2\}}.$$

By (3.18), we have that $d_{\{x_1,x_2\}} = f(\{x_1\})$, which results in $g(\{x_1, x_2\}) = f(\{x_1\}) + f(\{x_2\})$.

Now assume that the claim holds for $k \leq l$ $(l > 2)$. Then we prove that it also holds for $k = l + 1$ $(l > 2)$. Without loss of generality, we assume that $x_1 < x_2 < \cdots < x_{l+1}$. Then by (3.15) and the induction hypothesis for $k \leq l$, we have that

$$g(\{x_1, x_2, \ldots, x_{l+1}\}) = g(\{x_2, \ldots, x_{l+1}\}) + d = \sum_{j=2}^{l+1} f(\{x_j\}) + d.$$

For any $B = \{x_1, x_2, \ldots, x_{l+1}\} \setminus \{x_m\}$ and $a = x_n \in B$, by (3.18), we have that

$$\begin{aligned} d = \min_{m,n} \Bigg\{ &\sum_{j=1}^{l+1} f(\{x_j\}) - f(\{x_m\}) - \sum_{j=2}^{l+1} f(\{x_j\}) + \sum_{j=1}^{l+1} f(\{x_j\}) - f(\{x_n\}) \\ &- \left(\sum_{j=1}^{l+1} f(\{x_j\}) - f(\{x_m\}) - f(\{x_n\}) \right) \Bigg\} \\ = f(\{x_1\}), & \end{aligned}$$

which results in

$$g(\{x_1, x_2, \ldots, x_{l+1}\}) = \sum_{j=1}^{l+1} f(\{x_j\}).$$

This completes the proof. $\square$

Theorem 3.2.4 shows that any polymatroid function defined on the uniform matroid of rank 1 can be majorizingly extended to one defined on the whole power set. The following counterexample shows that the same is not the case for a uniform matroid of rank 2.

Example 2: Let $X = \{1, 2, 3, 4\}$ and $\mathcal{I} = \{A : A \in X \text{ and } |A| \leq 2\}$. Define $f : \mathcal{I} \to \mathbb{R}$ as follows:

$f(\emptyset) = 0,$

$f(\{1\}) = 1, f(\{2\}) = 2, f(\{3\}) = 3, f(\{4\}) = 4,$

$f(\{1, 2\}) = 2.0760, f(\{1, 3\}) = 3.2399, f(\{2, 3\}) = 3.3678,$

$f(\{1, 4\}) = 4.1233, f(\{2, 4\}) = 4.4799, f(\{3, 4\}) = 5.2518.$

It is easy to check that f is a polymatroid function on the uniform matroid of rank 2. Now we show that f can not be majorizingly extended to one on the whole power set. Let g denote the function obtained by the majorizing extension.

By (3.15) and (3.18), we have that $g(\{1,2,3\}) = f(\{2,3\}) + d_1$, and

$$\begin{aligned} d_1 &= \min\{f(\{1,2\} - f(\{2\}), f(\{1,2\}) - f(\{2,3\} + f(\{1,3\} - f(\{1\}), f(\{1,3\}) - f(\{3\})\} \\ &= 0.0760, \end{aligned}$$

which results in $g(\{1,2,3\}) = f(\{2,3\}) + d_1 = 3.4438$.

Similarly, we have that $g(\{1,2,4\}) = f(\{2,4\}) + d_2$, $g(\{1,3,4\}) = f(\{3,4\}) + d_3$, and $g(\{2,3,4\}) = f(\{3,4\}) + d_4$, where

$$\begin{aligned} d_2 &= \min\{f(\{1,2\}) - f(\{2\}), f(\{1,2\}) - f(\{2,4\}) + f(\{1,4\}) - f(\{1\}), f(\{1,4\}) - f(\{4\})\} \\ &= 0.0760, \end{aligned}$$

$$\begin{aligned} d_3 &= \min\{f(\{1,3\} - f(\{3\}), f(\{1,3\}) - f(\{3,4\}) + f(\{1,4\}) - f(\{1\}), f(\{1,4\}) - f(\{4\})\} \\ &= 0.1233, \end{aligned}$$

and

$$\begin{aligned} d_4 &= \min\{f(\{2,3\} - f(\{3\}), f(\{2,3\}) - f(\{3,4\}) + f(\{2,4\}) - f(\{2\}), f(\{2,4\}) - f(\{4\})\} \\ &= 0.3678. \end{aligned}$$

Hence, we have $g(\{1,2,4\}) = f(\{2,4\}) + d_2 = 4.5559$, $g(\{1,3,4\}) = f(\{3,4\}) + d_3 = 5.3751$, and $g(\{2,3,4\}) = f(\{3,4\}) + d_4 = 5.6196$.

Now majorizingly construct $g(\{1,2,3,4\})$. By (3.15) and (3.18), we have that $g(\{1,2,3,4\}) = g(\{2,3,4\}) + d_5$, and

$$
\begin{aligned}
d_5 = \min\{&g(\{1,2,3\}) - g(\{2,3,4\} + g(\{1,3,4\}) - f(\{1,3\}),\\
&g(\{1,2,3\}) - g(\{2,3,4\}) + g(\{1,2,4\}) - f(\{1,2\}),\\
&g(\{1,2,4\}) - g(\{2,3,4\}) + g(\{1,3,4\}) - f(\{1,4\}),\\
&g(\{1,2,3\} - f(\{2,3\}), g(\{1,2,4\}) - f(\{2,4\}), g(\{1,3,4\}) - f(\{3,4\})\}\\
= -0.0406 < 0.&
\end{aligned}
$$

Therefore, g defined as above is not a polymatroid function. However, there are some polymatroid functions defined on the uniform matroid of rank 2 that can be majorizingly extended to ones defined on the entire power set. In Section 3.4, we present two canonical examples that frequently arise in task scheduling and adaptive sensing and show that the objective functions in the two examples can be both majorizingly extended to polymatroid functions defined on the entire power set. Theorem 3.2.4 implies that any monotone additive function defined on the uniform matroid of rank k ($k > 1$) can be majorizingly extended to one defined on the entire power set.

3.2.4 Polymatroid Extension: From General Matroid to Power Set

Theorem 3.2.3 and Proposition 3.2.1 together provide an iterative algorithm for us to extend a polymatroid function f defined on the matroid $(X, \mathcal{I})$ to a polymatroid function g defined on the entire power set. We use g_k to denote a polymatroid function defined on the uniform matroid of rank k satisfying $g_k(A) = f(A)$ for $A \in \mathcal{I}$ with $|A| \leq k$. The idea is that we first define $g_1(A) = f(A)$ for $A \in \mathcal{I}$ with $|A| \leq 1$ and $g_1(A) \geq 0$ for $A \notin \mathcal{I}$ with $|A| = 1$. Then, iteratively extend g_k defined on the uniform matroid of rank k to g_{k+1} defined on the uniform matroid of rank $k+1$ using (3.15) and (3.17) for $k = 1, 2, \ldots, |X| - 1$. Finally, set $g = g_{|X|}$. This results in

$$
g_{k+1}(A) = \begin{cases} g_k(A), & |A| \leq k \\ f(A), & A \in \mathcal{I} \text{ with } |A| = k+1 \\ g_k(B^*) + d_A, & A \notin \mathcal{I} \text{ with } |A| = k+1. \end{cases}
$$

The specific process is given as follows:

1. First define
$$g_1(A) = \begin{cases} f(A), & A \in \mathcal{I} \text{ with } |A| \le 1 \\ d_A, & A \notin \mathcal{I} \text{ with } |A| = 1 \end{cases}$$
where $d_A \ge 0$.

2. Then iteratively define $g_{k+1}(A)$ for $k = 1, \dots, |X| - 1$ using the following method:

   Assume that $g_k(A)$ is well defined for $|A| \le k$. For $A \subseteq X$ with $|A| \le k$, set $g_{k+1}(A) = g_k(A)$. For $A \in \mathcal{I}$ with $|A| = k+1$, set $g_{k+1}(A) = f(A)$. For $A \notin \mathcal{I}$ with $|A| = k+1$, let
$$B^* \in \operatorname*{argmax}_{B: B \subseteq A, |B| = k} g_k(B).$$
   If
$$d^* = \min_{a \in B \subseteq A} \{[g_k(B) - g_k(B \setminus \{a\})] - [g_k(B^*) - g_k(A \setminus \{a\})]\} \ge 0,$$
   then set $g_{k+1}(A) = g_k(B^*) + d_A$, where $0 \le d_A \le d^*$; else, extension fails.

3. If $g_{|X|}$ exists, set $g = g_{|X|}$.

In the algorithm above, we do not specify the exact d_A value. Of course, as before, we can choose $d_A = d^*$, leading to a majorizing extension. As we have seen before, the majorizing extension might not be a polymatroid function even if a polymatroid extension exists. Nonetheless, if indeed a polymatroid extension exists, then there always exist choices of d_A that produce the extension via the algorithm above. But the problem of finding an appropriate sequence of d_A values can be reduced to that of finding a feasible path in a shortest-path problem (where shortest here could be defined in terms of the smallest total curvature of the extension). Solving this problem is tantamount to solving a problem of the form (1); in general, we would need to resort to something like dynamic programming. This implies that in general, finding a polymatroid function extension is nontrivial.

3.3 Improved Bounds

Let $f : 2^X \to \mathbb{R}$ be a polymatroid function. Note that $f : 2^X \to \mathbb{R}$ is itself an extension of f from $\mathcal{I}$ to the entire 2^X, and the extended $f : 2^X \to \mathbb{R}$ is a polymatroid function on the entire 2^X. Therefore, Theorem 3.2.3 gives rise in a straightforward way to the following result, stated without proof.

Proposition 3.3.1. *Let $(X, \mathcal{I})$ be a matroid and $f : 2^X \to \mathbb{R}$ a polymatroid function on 2^X. Then $c(f) \geq \inf_{g \in \Omega_f} c(g)$, where Ω_f is the set of all polymatroid functions g on 2^X that agree with f on $\mathcal{I}$.*

In this section, we will prove that for problem (3.1), if we set $d = \inf_{g \in \Omega_f} c(g)$, then the greedy strategy yields a $1/(1+d)$-approximation and a $(1-e^{-d})/d$-approximation under a general matroid and a uniform matroid constraint, respectively. Some proofs in this section are straightforward, but are included for completeness.

Theorem 3.3.2. *Let $(X, \mathcal{I})$ be a matroid of rank K and $f : \mathcal{I} \to \mathbb{R}$ a polymatroid function. If there exists an extension of f to the entire power set, then any greedy solution G to problem (3.1) satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{1+d}, \tag{3.19}$$

where $d = \inf_{g \in \Omega_f} c(g)$. In particular, when $(X, \mathcal{I})$ is a uniform matroid, any greedy solution G to problem (3.1) satisfies

$$\frac{f(G)}{f(O)} \geq \frac{1}{d}\left(1 - \left(1 - \frac{d}{K}\right)^K\right) > \frac{1}{d}\left(1 - e^{-d}\right). \tag{3.20}$$

Proof. By Theorems 3.1.1 and 3.1.2, for any extension g of f to the entire power set, we have the following inequalities

$$\frac{g(G)}{g(O)} \geq \frac{1}{1+c(g)}$$

and

$$\frac{g(G)}{g(O)} \geq \frac{1}{c(g)}\left(1 - \left(1 - \frac{c(g)}{K}\right)^K\right) > \frac{1}{c(g)}\left(1 - e^{-c(g)}\right).$$

Because f and g agree on $\mathcal{I}$, we have that $f(G) = g(G)$ and $f(O) = g(O)$. Thus, (3.19) and (3.20) hold for problem (3.1). $\square$

Remark 3.3.1. *Because the functions $1/(1+x)$, $(1-(1-x/K)^K)/x$, and $(1-e^{-x})/x$ are all nonincreasing in x for $x \in (0,1]$ and from Proposition 3.3.1 we have $0 < d \leq c(f) \leq 1$ when f is defined on the entire power set, we have that $1/(1+d) \geq 1/(1+c(f))$, $((1-(1-d/K)^K)/d \geq (1-(1-c(f)/K)^K)/c(f)$, and $(1-e^{-d})/d \geq (1-e^{-c(f)})/c(f)$. This implies that our new bounds are, in general, stronger than the previous bounds.*

Remark 3.3.2. *The bounds $1/(1+d)$ and $(1-e^{-d})/d$ apply to problems where the objective function is a polymatroid function defined only for sets in the matroid and can be extended to one defined on the entire power set. However, these bounds still depend on sets not in the matroid, because of the way d is defined.*

Now we define a notion of partial curvature that only involves sets in the matroid. Let $h : \mathcal{I} \to \mathbb{R}$ be a set function. We define the partial curvature $b(h)$ as follows:

$$b(h) = \max_{\substack{j,A: j \in A \in \mathcal{I} \\ h(\{j\}) \neq h(\emptyset)}} \left\{ 1 - \frac{h(A) - h(A \setminus \{j\})}{h(\{j\}) - h(\emptyset)} \right\}. \tag{3.21}$$

For convenience, we use b to denote $b(h)$ when there is no ambiguity. Note that $0 \leq b \leq 1$ when h is a polymatroid function on the matroid $(X, \mathcal{I})$, and $b = 0$ if and only if h is additive for sets in $\mathcal{I}$. When $b = 0$, the greedy solution to problem (3.1) coincides with the optimal solution, so we only consider $b \in (0,1]$ in the rest of the paper. For any extension of $f : \mathcal{I} \to \mathbb{R}$ to $g : 2^X \to \mathbb{R}$, we have that $c(g) \geq b(f)$, which will be proved in the following theorem.

Theorem 3.3.3. *Let $(X, \mathcal{I})$ be a matroid and $f : \mathcal{I} \to \mathbb{R}$ a polymatroid function. Assume that a polymatroid extension $g : 2^X \to \mathbb{R}$ of f exists. Then $b(f) \leq c(g)$.*

Proof. By submodularity of g and $g(A) = f(A)$ for any $j \in A \in \mathcal{I}$, we have that

$$f(A) - f(A \setminus \{j\}) \geq g(X) - g(X \setminus \{j\}),$$

which implies that for any $j \in A \in \mathcal{I}$,

$$1 - \frac{f(A) - f(A \setminus \{j\})}{f(\{j\}) - f(\emptyset)} \leq 1 - \frac{g(X) - g(X \setminus \{j\})}{g(\{j\}) - g(\emptyset)}.$$

Hence, combining the above with (3.2) and (3.21) gives $b(f) \leq c(g)$. $\square$

Remark 3.3.3. *As mentioned earlier, the improved bounds involving d in Theorem 3.3.2 still depend on sets not in the matroid. In contrast, by definition, the partial curvature $b(f)$ depends on sets in the matroid. So if there exists an extension of f to g such that $c(g) = b(f)$, then we can derive bounds that are not influenced by sets outside the matroid. However, it turns out that there does not always exist a g such that $c(g) = b(f)$; we will give an example in Section 4.2 to show this. In the following theorem, we provide necessary and sufficient conditions for $c(g) = b(f)$.*

Theorem 3.3.4. *Let $(X, \mathcal{I})$ be a matroid and $f : \mathcal{I} \to \mathbb{R}$ a polymatroid function. Let $g : 2^X \to \mathbb{R}$ be a polymatroid function that agrees with f on $\mathcal{I}$. Then $c(g) = b(f)$ if and only if*

$$g(X) - g(X \setminus \{a\}) \geq (1 - b(f))g(\{a\}) \tag{3.22}$$

for any $a \in X$, and equality holds for some $a \in X$.

Proof. $\rightarrow$

In this direction, we assume that $c(g) = b(f)$ and then to prove that $g(X) - g(X \setminus \{a\}) \geq (1 - b(f))g(\{a\})$ for any $a \in X$ and that equality holds for some $a \in X$. By the definition of the total curvature c of g and $c(g) = b(f)$, we have for any $a \in X$,

$$g(X) - g(X \setminus \{a\}) \geq (1 - b(f))g(\{a\}),$$

and equality holds for some $a \in X$. $\leftarrow$

Now we assume that $g(X) - g(X \setminus \{a\}) \geq (1 - b(f))g(\{a\})$ for any $a \in X$ and that equality holds for some $a \in X$, and then prove that $c(g) = b(f)$. By the assumptions, we have

$$1 - \frac{g(X) - g(X \setminus \{a\})}{g(\{a\}) - g(\emptyset)} \leq b(f)$$

for any $a \in X$, and equality holds for some $a \in X$. By the definition of the total curvature c of g, we have

$$c(g) = \max_{\substack{a \in X \\ g(\{a\}) \neq g(\emptyset)}} \left\{ 1 - \frac{g(X) - g(X \setminus \{a\})}{g(\{a\}) - g(\emptyset)} \right\} = b(f).$$

This completes the proof. □

Remark 3.3.4. *In Section 3.4, we will provide a task scheduling example to show that there exists a polymatroid function $g : 2^X \to \mathbb{R}$ that agrees with $f : \mathcal{I} \to \mathbb{R}$ such that $c(g) = b(f)$. We also provide a contrasting example from an adaptive sensing problem where such an extension does not exist.*

Combining Theorems 3.3.2 and 3.3.4, we have the following corollary.

Corollary 3.3.5. *Let $(X, \mathcal{I})$ be a matroid of rank K. Let $g : 2^X \to \mathbb{R}$ be a polymatroid function that agrees with f on $\mathcal{I}$ and $g(X) - g(X \setminus \{a\}) \geq (1 - b(f))g(\{a\})$ for any $a \in X$ with equality holding for some $a \in X$. Then, any greedy solution G to problem* (3.1) *satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{1 + b(f)}. \tag{3.23}$$

In particular, when $(X, \mathcal{I})$ is a uniform matroid, any greedy solution G to problem (3.1) *satisfies*

$$\frac{f(G)}{f(O)} \geq \frac{1}{b(f)} \left(1 - \left(1 - \frac{b(f)}{K} \right)^K \right) > \frac{1}{b(f)} \left(1 - e^{-b(f)} \right). \tag{3.24}$$

The bounds $1/(1 + b(f))$ and $(1 - (1 - b(f)/K)^K)/b(f)$ do not depend on sets outside the matroid, so they apply to problems where the objective function is only defined on the matroid, provided that an extension that satisfies the assumptions in Theorem 3.3.4 exists. When f is defined on the entire power set, from Theorem 3.3.3, we have $b(f) \leq c(f)$, which implies that the bounds are stronger than those from [34].

3.4 Examples

We first provide a task scheduling example where we majorizingly extend $f : \mathcal{I} \to \mathbb{R}$ to a polymatroid function $g_1 : 2^X \to \mathbb{R}$ with $c(g_1) > b(f)$. We also extend $f : \mathcal{I} \to \mathbb{R}$ to another polymatroid function $g_2 : 2^X \to \mathbb{R}$ with $c(g_2) = b(f)$. The two extensions both result in stronger bounds than the previous bound from [34]. Then we provide an adaptive sensing example to majorizingly extend $f : \mathcal{I} \to \mathbb{R}$ to a polymatroid function $g_1 : 2^X \to \mathbb{R}$ and show that there does not exist any extension of f to g such that $c(g) = b(f)$ holds. However, in this example, it turns out that for our majorizing extension g_1, $c(g_1)$ is very close to $b(f)$ and is much smaller than $c(f)$.

3.4.1 Task Scheduling

As a canonical example of problem (3.1), we will consider the task assignment problem that was posed in [3], and was further analyzed in [4, 53]. In this problem, there are n subtasks and a set X of N agents a_j $(j = 1, \ldots, N)$. At each stage, a subtask i is assigned to an agent a_j, who successfully accomplishes the task with probability $p_i(a_j)$. Let $X_i(a_1, a_2, \ldots, a_k)$ denote the Bernoulli random variable that describes whether or not subtask i has been accomplished after performing the sequence of actions $a_1, a_2, \ldots, a_k$ over k stages. Then $\frac{1}{n}\sum_{i=1}^{n} X_i(a_1, a_2, \ldots, a_k)$ is the fraction of subtasks accomplished after k stages by employing agents $a_1, a_2, \ldots, a_k$. The objective function f for this problem is the expected value of this fraction, which can be written as

$$f(\{a_1, \ldots, a_k\}) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \prod_{j=1}^{k}(1 - p_i(a_j))\right).$$

Assume that $p_i(a) > 0$ for any $a \in X$; then it is easy to check that f is non-decreasing. Therefore, when $\mathcal{I} = \{S \subseteq X : |S| \leq K\}$, the solution to this problem should be of size K. Also, it is easy to check that the function f has the submodular property.

For convenience, we only consider the special case $n = 1$; our analysis can be generalized to any $n \geq 2$. In this case, we have

$$f(\{a_1, \dots, a_k\}) = 1 - \prod_{j=1}^{k} (1 - p(a_j)), \tag{3.25}$$

where $p(\cdot) = p_1(\cdot)$.

Let $X = \{a_1, a_2, a_3, a_4\}$, $p(a_1) = 0.4$, $p(a_2) = 0.6$, $p(a_3) = 0.8$, and $p(a_4) = 0.9$. Then, $f(A)$ is defined as in (3.25) for any $A = \{a_i, \dots, a_k\} \subseteq X$. Consider $K = 2$, then $\mathcal{I} = \{S \subseteq X : |S| \leq 2\}$. It is easy to show that $f : \mathcal{I} \to \mathbb{R}$ is a polymatroid function. We first majorizingly extend $f : \mathcal{I} \to \mathbb{R}$ to a polymatroid function $g_1 : 2^X \to \mathbb{R}$ using (3.15) and (3.18). By (3.15), we have that $g_1(\{a_1, a_2, a_3\}) = f(\{a_2, a_3\}) + d_{\{a_1,a_2,a_3\}}$, $g_1(\{a_1, a_2, a_4\}) = f(\{a_2, a_4\}) + d_{\{a_1,a_2,a_4\}}$, $g_1(\{a_1, a_3, a_4\}) = f(\{a_3, a_4\}) + d_{\{a_1,a_3,a_4\}}$, and $g_1(\{a_2, a_3, a_4\}) = f(\{a_3, a_4\}) + d_{\{a_2,a_3,a_4\}}$. By (3.18), we have that

$$\begin{aligned}
d_{\{a_1,a_2,a_3\}} &= \min\{f(\{1,2\} - f(\{2\}), f(\{1,2\}) - f(\{2,3\} + f(\{1,3\} - f(\{1\}), \\
&\qquad f(\{1,3\}) - f(\{3\})\} \\
&= 0.08,
\end{aligned}$$

$$\begin{aligned}
d_{\{a_1,a_2,a_4\}} &= \min\{f(\{1,2\}) - f(\{2\}), f(\{1,2\}) - f(\{2,4\}) + f(\{1,4\}) - f(\{1\}), \\
&\qquad f(\{1,4\}) - f(\{4\})\} \\
&= 0.04,
\end{aligned}$$

$$\begin{aligned}
d_{\{a_1,a_3,a_4\}} &= \min\{f(\{1,3\} - f(\{3\}), f(\{1,3\}) - f(\{3,4\}) + f(\{1,4\}) - f(\{1\}), \\
&\qquad f(\{1,4\}) - f(\{4\})\} \\
&= 0.04,
\end{aligned}$$

$$
\begin{aligned}
d_{\{a_2,a_3,a_4\}} &= \min\{f(\{2,3\} - f(\{3\}), f(\{2,3\}) - f(\{3,4\}) + f(\{2,4\}) - f(\{2\}),\\
&\qquad f(\{2,4\}) - f(\{4\})\}\\
&= 0.06.
\end{aligned}
$$

Hence, $g_1(\{a_1, a_2, a_3\}) = 1, g_1(\{a_1, a_2, a_4\}) = 1, g_1(\{a_1, a_3, a_4\}) = 1.02$, and $g_1(\{a_2, a_3, a_4\}) = 1.04$.

We now construct $g_1(X)$. By (3.15), we have that $g_1(X) = g_1(\{a_2, a_3, a_4\}) + d_X$. By (3.18), we have that

$$
\begin{aligned}
d_X = \min\{&g_1(\{1,2,3\}) - g_1(\{2,3,4\} + g_1(\{1,3,4\}) - f(\{1,3\}),\\
&g_1(\{1,2,3\}) - g_1(\{2,3,4\}) + g_1(\{1,2,4\}) - f(\{1,2\}),\\
&g_1(\{1,2,4\}) - g_1(\{2,3,4\}) + g_1(\{1,3,4\}) - f(\{1,4\})\\
&g_1(\{1,2,3\} - f(\{2,3\}), g_1(\{1,2,4\}) - f(\{2,4\}), g_1(\{1,3,4\}) - f(\{3,4\})\}\\
= 0.04,&
\end{aligned}
$$

Hence, $g_1(X) = g_1(\{a_2, a_3, a_4\}) + d_X = 1.08$. Therefore, g_1 defined as above is a majorizing extension of f from $\mathcal{I}$ to the whole power set.

The total curvature c of $g_1 : 2^X \to \mathbb{R}$ is

$$
\begin{aligned}
c(g_1) &= \max_{a_i \in X} \left\{ 1 - \frac{g(X) - g(X \setminus \{a_i\})}{g(\{a_i\}) - g(\emptyset)} \right\}\\
&= 0.911.
\end{aligned}
$$

In contrast, the total curvature c of f is

$$
\begin{aligned}
c(f) &= \max_{a_i \in X} \left\{ 1 - \frac{f(X) - f(X \setminus \{a_i\})}{f(\{a_i\}) - f(\emptyset)} \right\} \\
&= \max_{a_i, a_j, a_k \in X} \{1 - (1 - p(\{a_i\}))\,(1 - p(\{a_j\}))(1 - p(\{a_k\}))\} \\
&= 0.992.
\end{aligned}
$$

By the definition of the partial curvature b of f, we have

$$
\begin{aligned}
b(f) &= \max_{\substack{j \in A \subseteq X, |A|=2, \\ f(\{j\}) \neq 0}} \left\{ 1 - \frac{f(A) - f(A \setminus \{j\})}{f(\{j\}) - f(\emptyset)} \right\} \\
&= \max_{\{a_i, a_j\} \subseteq X} \left\{ 1 - \frac{f(\{a_i, a_j\}) - f(\{a_i\})}{f(\{a_j\})} \right\} \\
&= \max_{a_i \in X} \{p(\{a_i\})\} \\
&= 0.9.
\end{aligned}
$$

We can see that $c(g_1)$ is close to $b(f)$ and smaller than $c(f)$ though $c(g_1) \neq b(f)$.

Next, we give another extension g_2 which satisfies that $c(g_2) = b(f)$. By (3.15), we have that $g_2(\{a_1, a_2, a_3\}) = f(\{a_2, a_3\}) + d_{\{a_1, a_2, a_3\}}$, $g_2(\{a_1, a_2, a_4\}) = f(\{a_2, a_4\}) + d_{\{a_1, a_2, a_4\}}$, $g_2(\{a_1, a_3, a_4\}) = f(\{a_3, a_4\}) + d_{\{a_1, a_3, a_4\}}$, and $g_2(\{a_2, a_3, a_4\}) = f(\{a_3, a_4\}) + d_{\{a_2, a_3, a_4\}}$. First, we will define $d_{\{a_1, a_2, a_3\}}$. By (3.17), we have that

$$
\begin{aligned}
d_{\{a_1, a_2, a_3\}} &\leq \min\{f(\{a_1, a_2\}) - f(\{a_2\}), f(\{a_1, a_3\}) - f(\{a_3\}), \\
&\qquad f(\{a_1, a_2\}) - f(\{a_2, a_3\}) + f(\{a_1, a_3\}) - f(\{a_1\})\} \\
&= 0.08.
\end{aligned}
$$

By (3.22), it suffices to have that

$$\begin{aligned}d_{\{a_1,a_2,a_3\}} \geq \max\{&(1-b)f(\{a_1\}), f(\{a_1,a_3\}) - f(\{a_2,a_3\}) + (1-b)f(\{a_2\}),\\ &f(\{a_1,a_2\}) - f(\{a_2,a_3\}) + (1-b)f(\{a_3\})\}\\ = 0.04.&\end{aligned}$$

Setting $d_{\{a_1,a_2,a_3\}} = 0.04$ to satisfy the above two inequalities gives that $g_2(\{a_1,a_2,a_3\}) = f(\{a_2,a_3\}) + d_{\{a_1,a_2,a_3\}} = 0.96$. Similarly, we set

$$\begin{aligned}g_2(\{a_1,a_2,a_4\}) &= f(\{a_2,a_4\}) + d_{\{a_1,a_2,a_4\}} = 1,\\ g_2(\{a_1,a_3,a_4\}) &= f(\{a_3,a_4\}) + d_{\{a_1,a_3,a_4\}} = 1.02,\\ g_2(\{a_2,a_3,a_4\}) &= f(\{a_3,a_4\}) + d_{\{a_2,a_3,a_4\}} = 1.04.\end{aligned}$$

We now define $g_2(X)$. By (3.15), we have that $g_2(X) = g_2(\{a_2,a_3,a_4\}) + d_X$. By (3.17), it suffices to have that

$$\begin{aligned}d_X \leq \min\{&g_2(\{a_1,a_2,a_4\}) - f(\{a_2,a_4\}),\\ &g_2(\{a_1,a_3,a_4\}) - f(\{a_3,a_4\}), g_2(\{a_1,a_2,a_3\}) - f(\{a_2,a_3\}),\\ &g_2(\{a_1,a_2,a_3\}) - g_2(\{a_2,a_3,a_4\}) + g_2(\{a_1,a_2,a_4\}) - f(\{a_1,a_2\}),\\ &g_2(\{a_1,a_3,a_4\}) - g_2(\{a_2,a_3,a_4\}) + g_2(\{a_1,a_2,a_4\}) - f(\{a_1,a_4\}),\\ &g_2(\{a_1,a_3,a_4\}) - g_2(\{a_2,a_3,a_4\}) + g_2(\{a_1,a_2,a_3\}) - f(\{a_1,a_3\})\}\\ = 0.04.&\end{aligned}$$

By (3.22), it suffices to have that

$$\begin{aligned}d_X \geq \max\{&(1-b)f(\{a_1\}),\\ &g_2(\{a_1,a_3,a_4\}) - g_2(\{a_2,a_3,a_4\}) + (1-b)f(\{a_2\}),\\ &g_2(\{a_1,a_2,a_4\}) - g_2(\{a_2,a_3,a_4\}) + (1-b)f(\{a_3\}),\\ &g_2(\{a_1,a_2,a_3\}) - g_2(\{a_2,a_3,a_4\}) + (1-b)f(\{a_4\})\}\\ = 0.04.&\end{aligned}$$

Setting $d_X = 0.04$ to satisfy the above two inequalities gives us $g_2(X) = g_2(\{a_2, a_3, a_4\}) + d_X = 1.08$.

The total curvature c of $g_2 : 2^X \to \mathbb{R}$ is

$$c(g_2) = \max_{a_i \in X} \left\{ 1 - \frac{g(X) - g(X \setminus \{a_i\})}{g(\{a_i\}) - g(\emptyset)} \right\} = 0.9 = b(f) < c(f) = 0.992.$$

By Corollary 3.3.5, we have that the greedy strategy for the task scheduling problem satisfies the bound $(1 - (1 - b(f)/2)^2)/b(f) = 0.775$, which is better than the previous bound $(1 - (1 - c(f)/2)^2)/c(f) = 0.752$.

3.4.2 Adaptive Sensing

For our second example, we consider the adaptive sensing design problem posed in [4, 53]. Consider a signal of interest $x \in \mathbb{R}^2$ with normal prior distribution $\mathcal{N}(0, I)$, where I is the 2×2 identity matrix; our analysis easily generalizes to dimensions larger than 2. Let $\mathbb{A} = \{\text{Diag}(\sqrt{\alpha}, \sqrt{1-\alpha}) : \alpha \in \{\alpha_1, \ldots, \alpha_N\}\}$, where $\alpha \in [0.5, 1]$ for $1 \leq i \leq N$. At each stage i, we make a measurement y_i of the form

$$y_i = a_i x + w_i,$$

where $a_i \in \mathbb{A}$ and w_i represents i.i.d. Gaussian measurement noise with mean zero and covariance I, independent of x.

The objective function f for this problem is the information gain [30], which can be written as

$$f(\{a_1, \ldots, a_k\}) = H_0 - H_k. \tag{3.26}$$

Here, $H_0 = \frac{N}{2}\log(2\pi e)$ is the entropy of the prior distribution of x and H_k is the entropy of the posterior distribution of x given $\{y_i\}_{i=1}^k$; that is,

$$H_k = \frac{1}{2}\log\det(P_k) + \frac{N}{2}\log(2\pi e),$$

where $P_k = \left(P_{k-1}^{-1} + a_k^T a_k\right)^{-1}$ is the posterior covariance of x given $\{y_i\}_{i=1}^k$.

The objective is to choose a set of measurement matrices $\{a_i^*\}_{i=1}^K$, $a_i^* \in \mathbb{A}$, to maximize the information gain $f(\{a_1, \ldots, a_K\}) = H_0 - H_K$. It is easy to check that f is monotone, submodular, and $f(\emptyset) = 0$; i.e., f is a polymatroid function.

Let $X = \{a_1, a_2, a_3\}$, $\alpha_1 = 0.5$, $\alpha_2 = 0.6$, and $\alpha_3 = 0.8$. Then, $f(A)$ is defined as in (3.26) for any $A = \{a_i, \ldots, a_k\} \subseteq X$. Consider $K = 2$, where $\mathcal{I} = \{S \subseteq X : |S| \leq 2\}$.

The total curvature of f is

$$\begin{aligned} c(f) &= \max_{a_i \in X} \left\{ 1 - \frac{f(X) - f(X \setminus \{a_i\})}{f(\{a_i\}) - f(\emptyset)} \right\} \\ &= 0.4509. \end{aligned}$$

We first majorizingly extend $f : \mathcal{I} \to \mathbb{R}$ to a polymatroid function g_1 defined on the whole power set. Then we show that there does not exist a polymatroid extension g_2 such that $c(g_2) = b(f)$. However, for the majorizing extension g_1, it turns out that $c(g_1)$ is very close to $b(f)$ and is much smaller than $c(f)$.

We start by majorizingly extending f to g_1. By (3.15) and (3.18), we have $g_1(X) = f(\{a_1, a_2\}) + d_X$, where

$$\begin{aligned} d_X = \min\{&f(\{a_1, a_3\}) - f(\{a_1\}), f(\{a_2, a_3\}) - f(\{a_2\}), \\ &f(\{a_1, a_3\}) - f(\{a_1, a_2\}) + f(\{a_2, a_3\}) - f(\{a_3\})\} \\ = \log &\sqrt{1.6799}. \end{aligned}$$

Hence, $g_1(X) = \log \sqrt{6.7028}$.

The total curvature of g_1 is

$$\begin{aligned} c(g_1) &= \max_{a_i \in X} \left\{ 1 - \frac{g_1(X) - g_1(X \setminus \{a_i\})}{g_1(\{a_i\}) - g_1(\emptyset)} \right\} \\ &= 0.3317. \end{aligned}$$

By the definition of the partial curvature b of f, we have

$$
\begin{aligned}
b(f) &= \max_{\substack{j \in A \subseteq X, |A|=2, \\ f(\{j\}) \neq 0}} \left\{ 1 - \frac{f(A) - f(A \setminus \{j\})}{f(\{j\}) - f(\emptyset)} \right\} \\
&= \max_{\{a_i, a_j\} \subseteq X} \left\{ 1 - \frac{f(\{a_i, a_j\}) - f(\{a_i\})}{f(\{a_j\})} \right\} \\
&= 0.3001.
\end{aligned}
$$

Comparing the values of $c(g_1), c(f)$, and $b(f)$, we have that $c(g_1)$ is much smaller than $c(f)$ and very close to $b(f)$. By Theorem 3.3.2, we have that the greedy strategy for the adaptive sensing problem satisfies the bound $(1-(1-c(g_1)/2)^2)/c(g_1) = 0.9172$, which is stronger than the previous bound $(1-(1-c(f)/2)^2)/c(f) = 0.8873$. Now we try to extend f to a polymatroid function g_2 such that $c(g_2) = b(f)$. By (3.15), $g_2(X) = f(\{a_1, a_2\}) + d_X$. By (3.17), it suffices to have that

$$
\begin{aligned}
d_X \leq \min\{&f(\{a_1, a_3\}) - f(\{a_1\}), f(\{a_2, a_3\}) - f(\{a_2\}), \\
&f(\{a_1, a_3\}) - f(\{a_1, a_2\}) + f(\{a_2, a_3\}) - f(\{a_3\})\} \\
= \log &\sqrt{1.6799}.
\end{aligned}
$$

By (3.22), it suffices to have that

$$
\begin{aligned}
d_X \geq \max\{&(1-b(f))f(\{a_3\}), \\
&f(\{a_2, a_3\}) - f(\{a_1, a_2\}) + (1-b(f))f(\{a_1\}), \\
&f(\{a_1, a_3\}) - f(\{a_1, a_2\}) + (1-b(f))f(\{a_2\})\}, \\
= \log &\sqrt{1.7232}.
\end{aligned}
$$

Comparing the above two inequalities, we see that there does not exist d_X such that g_2 is a polymatroid function satisfying $c(g_2) = b(f)$.

Chapter 4

Performance of Nash Equilibria in Utility Systems

In this chapter, we use similar techniques for bounding the batched greedy strategy in Chapter 2 to bound the performance of Nash equilibria when there is some notion of "grouping" among users in noncooperative games. The connection to the game setting is associating our objective function with a social utility function, greedy strategies with Nash equilibria, and batching with cooperation of subgroups in games.

We consider two notions of grouping that yield to provable performance bounds. The first type of grouping we consider is the recent framework of [2], where associated with each user is a private objective function and a fixed group of users having some social ties with it. Each user's strategy maximizes an objective function called the social group utility, which is the sum of its private objective function and a linear combination of the private objective functions of users in its group. Within this setting, [2] define what they call a social-aware Nash equilibrium, where no user can improve its social group utility by unilaterally changing its strategy. We will show that this framework yields to the bounding results of [1] for noncooperative games, thus establishing provable performance guarantees for the framework of [2].

In the second type of grouping we consider, the set of users is partitioned into disjoint groups. Associated with each group is a group utility function. Users within a group cooperate in the sense that their strategy is to (jointly) maximize the group utility function, giving rise to a natural definition of group Nash equilibrium. Although we can view each group as a new user with vector-valued actions so that a similar $1/2$ bound to the result of [1] holds, we would like to investigate the performance bound for the group Nash equilibrium in terms of curvature and compare it with the case where there is no grouping. We define a measure of group curvature and derive an associated lower bound involving this curvature. We prove that this bound is tighter than that for the case without grouping among users, accounting for the cooperation within the groups. We also prove that, under the condition that each user has the same action space, the higher the degree of cooperation, the tighter the lower bound.

This chapter is organized as follows. In Section 4.1, we introduce our notation and some definitions that will be used throughout the paper. In Section 4.2, we review the bounding results of [1]. In Section 4.3, we first describe the framework of [2] and show that a social-aware utility system yields to the bounding results of Vetta for non-cooperative system, thus establishing provable performance guarantees for the social-aware Nash equilibrium. Next, we describe our second type of grouping involving l disjoint groups with in-group cooperation. In this case, each group can be viewed as a new user with vector-valued actions, and a 1/2 bound for the performance of group Nash equilibrium follows from the result of [1]. We then define the group curvature c_{k_i} associated with group i with k_i users, and we show that if the social utility function is nondecreasing and submodular, then any group Nash equilibrium achieves at least $1/(1+\max_{1\leq i\leq l} c_{k_i})$ of the optimal social utility, which is tighter than that for the case without grouping. Especially, if each user has the same action space, then we have that any group Nash equilibrium achieves at least $1/(1+c_{k^*})$ of the optimal social utility, where k^* is the least number of users among all the groups. In Section 4.4, we present an example of a utility system for database assisted spectrum access, adopted from [2]. We show that the utility system for this example is valid and the social utility function is submodular, illustrating an application of our results.

The results in this chapter were published in [61].

4.1 Preliminaries

In this section, we first introduce notation and a number of definitions used throughout the paper.

4.1.1 Actions

Suppose we have a set $\mathcal{N} = \{1, 2, \dots, N\}$ of N users and ground sets $V_1, V_2, \dots, V_N$, where each element in V_i denotes an act that user i can take. We call a set of acts an action, and if an action

$x_i \subseteq V_i$ is available to user i we call it a feasible action. We denote by $\mathcal{X}_i$ the set of all feasible actions for user i, i.e., $\mathcal{X}_i = \{x_i \subseteq V_i : x_i \text{ is a feasible action}\}$, with $n_i = |\mathcal{X}_i|$ the cardinality of $\mathcal{X}_i$.

Let $\mathcal{X} = \prod_{i=1}^{N} \mathcal{X}_i$ and $X = (x_{i_1}, \ldots, x_{i_k})$, where $x_j \in \mathcal{X}_j$, with $i_1 \leq j \leq i_k$. We call X an action sequence of length k in $\mathcal{X}$. This sequence includes the actions taken by users $i_1, \ldots, i_k$ in order. Given an action sequence X, suppose Y is formed by removing some of the elements of X without changing the order of the remaining elements. Then, we call the derived action sequence Y a subsequence of X and denote this relation by $Y \subseteq X$. This follows the definition of a subsequence in [62].

Consider an action sequence $X = (x_1, \ldots, x_N) \in \mathcal{X}$. Then, $X_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$ is the subsequence of X that includes actions taken by all users except user i. We use (X_{-i}, x_i') to denote the action sequence $(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_N)$ that results from X when user i changes its action from x_i to x_i'.

Given action sequences $Y = (y_{i_1}, \ldots, y_{i_k})$ and $Z = (z_{j_1}, \ldots, z_{j_l})$, we define $Y \oplus Z = (y_{i_1}, \ldots, y_{i_k}, z_{j_1}, \ldots, z_{j_l})$ as the concatenation of Y and Z when $i_p \neq j_q$ for $1 \leq p \leq k$ and $1 \leq q \leq l$ (following the notation in [4]).

4.1.2 Strategies

Let $s_i = (s_i^1, \ldots, s_i^{n_i})$, where $s_i^j \geq 0$ is the probability with which user i takes action j and $\sum_{j=1}^{n_i} s_i^j = 1$. Following the terminology of [1], we call s_i a strategy taken by user i. When $s_i^j = 1$ and $s_i^l = 0$ for $1 \leq j \leq n_i$ and $l \neq j$, we say that user i takes a pure strategy. Otherwise, we say that user i takes a mixed strategy.

Let $\mathcal{S}_i = \{s_i \in \mathbb{R}_i^{n_i} : \sum_{j=1}^{n_i} s_i^j = 1, s_i^j \geq 0\}$ be the strategy space for user i and $\mathcal{S} = \prod_{i=1}^{N} \mathcal{S}_i$. Similar to the definition of an action sequence, we call $S = (s_{i_1}, \ldots, s_{i_k})$, with $s_j \in \mathcal{S}_j$ and $i_1 \leq j \leq i_k$, a strategy sequence of length k in $\mathcal{S}$. Then a subsequence T of S is a sequence derived from S by deleting some elements without changing the order of the remaining elements. We define $S_i = (s_1, \ldots, s_i)$, for $1 \leq i \leq N$, as a sequence of strategies taken by users $1, \ldots, i$.

Given a strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$, the sequence $S_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N)$ is the subsequence of S that contains strategies taken by all users except user i, and $(S_{-i}, s'_i) = (s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_N)$ is the strategy sequence that results from S when user i changes its strategy from s_i to s'_i.

Given strategy sequences $T = (t_{i_1}, \ldots, t_{i_k})$ and $W = (w_{j_1}, \ldots, w_{j_l})$, we write $T \oplus W = (t_{i_1}, \ldots, t_{i_k}, w_{j_1}, \ldots, w_{j_l})$ for the concatenation of T and W when $i_p \neq j_q$ for $1 \leq p \leq k$ and $1 \leq q \leq l$.

4.1.3 Utility Functions

We define the social utility function as a mapping γ from sequences in $\mathcal{X}$ to real numbers, and the private utility function for user i $(1 \leq i \leq N)$ as a mapping α_i from sequences in $\mathcal{X}$ to real numbers. Correspondingly, we define $\bar{\gamma}$ and $\bar{\alpha}_i$ as mappings, from sequences in $\mathcal{S}$ to real numbers, that correspond to the expectations of γ and α_i, respectively. We call $\bar{\gamma}$ the expected social utility function and $\bar{\alpha}_i$ the expected private utility function for user i. We also define $\gamma_Z(Y) = \gamma(Y \oplus Z) - \gamma(Y)$ for any Y, Z in $\mathcal{X}$ such that $Y \oplus Z$ is well defined, and $\bar{\gamma}_W(T) = \bar{\gamma}(T \oplus W) - \bar{\gamma}(T)$ for any T, W in $\mathcal{S}$ such that $T \oplus W$ is defined.

We denote by Ω the optimal sequence of strategies in maximizing an expected utility function $\bar{\gamma}$, and assume that $\Omega = (\sigma_1, \ldots, \sigma_N)$ is composed of pure strategies $\sigma_i \in \mathcal{S}_i$, $i = 1, \ldots, N$. For convenience, we also use σ_i to denote the optimal action that user i takes. Then, we have that the optimal value of $\bar{\gamma}$, denoted by OPT, is OPT $= \bar{\gamma}(\Omega) = \gamma(\Omega)$.

4.1.4 Curvature, Monotoneity, and Submodularity

Given a strategy sequence $S_i = (s_1, \ldots, s_i)$ for $1 \leq i \leq N$, we use the notation $\Omega \cup S_i$ to represent the sequence in which user j $(1 \leq j \leq i)$ implements the actions $\sigma_j \cup x_j^1, \ldots, \sigma_j \cup x_j^{n_j}$ with probabilities $s_j^1, \ldots, s_j^{n_j}$, and user j $(j > i)$ plays the action σ_j, so $\bar{\gamma}(\Omega \cup S_i)$ is well defined. Then the curvature c of the expected social utility function $\bar{\gamma}$ is defined as

$$c = \max_{i:\bar{\gamma}_{s_i}(\emptyset)\neq 0} \left\{ 1 - \frac{\bar{\gamma}_{s_i}(\Omega \cup S_{-i})}{\bar{\gamma}_{s_i}(\emptyset)} \right\}.$$

The social utility function γ is called nondecreasing if for all subsequences Y of a sequence X in $\mathcal{X}$, i.e., $Y \subseteq X$ in $\mathcal{X}$, $f(Y) \leq f(X)$. It is called submodular if for all $Y \subseteq X$ and Z in $\mathcal{X}$ such that $X \oplus Z$ is defined, we have $\gamma_Z(Y) \geq \gamma_Z(X)$. Our terminology here is consistent with that of [62]. Because $\bar{\gamma}$ is the expected value of γ, we have that if γ is nondecreasing and submodular, then $\bar{\gamma}$ is also nondecreasing and submodular, respectively. So in the following sections, when we say that γ is nondecreasing and submodular, it implies that $\bar{\gamma}$ is nondecreasing and submodular, respectively.

4.2 Performance Bounds for Nash Equilibria

In this section, we first review the definitions of a Nash equilibrium and a valid utility system from [1]. We then review the bounds derived in [1] for the performance of any Nash equilibrium.

Definition 4.2.1. *A strategy sequence $S \in \mathcal{S}$ is a Nash equilibrium if no user has an incentive to unilaterally change its strategy, i.e., for any user i,*

$$\bar{\alpha}_i(S) \geq \bar{\alpha}_i((S_{-i}, s_i')), \quad \forall s_i' \in \mathcal{S}_i. \tag{4.1}$$

Assumption 4.2.1. *[1] The private utility of user i ($1 \leq i \leq N$) is at least as large as the loss in the social utility resulting from user i dropping out of the game. That is, the system ($\bar{\gamma}, \{\bar{\alpha}_i\}_{i=1}^N$) has the property that for any strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$,*

$$\bar{\alpha}_i(S) \geq \bar{\gamma}_{s_i}(S_{-i}), \quad \forall 1 \leq i \leq N. \tag{4.2}$$

Assumption 4.2.2. *[1] The sum of the private utilities of the system is not larger than the social utility, i.e., for any strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$,*

$$\sum_{i=1}^{N} \bar{\alpha}_i(S) \leq \bar{\gamma}(S). \tag{4.3}$$

A utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$ satisfying Assumptions 4.2.1 and 4.2.2 is called a valid system. Given $X \in \mathcal{X}$, if for any $1 \leq i \leq N$, the inequalities $\alpha_i(X) \geq \gamma_{x_i}(X_{-i})$ and $\sum_{i=1}^N \alpha_i(X) \leq \gamma(X)$ hold, then the inequalities (4.2) and (4.3) hold.

Theorem 4.2.1. *[1] For a valid utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$, if the social utility function γ is submodular, then for any Nash equilibrium $S \in \mathcal{S}$ we have*

$$\bar{\gamma}(S) \geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + \sum_{i=1}^{N} \bar{\gamma}_{s_i}(S_{-i} \cup \Omega)\right). \tag{4.4}$$

If γ is non-decreasing, then $\bar{\gamma}_{s_i}(S_{-i} \cup \Omega) \geq 0$ and the above inequality shows that any Nash equilibrium achieves at least $1/2$ of the optimal social utility function value.

Theorem 4.2.2. *[1] For a valid utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$, if the social utility function γ is non-decreasing and submodular, then for any Nash equilibrium $S \in \mathcal{S}$ we have*

$$\bar{\gamma}(S) \geq \frac{1}{1+c}\bar{\gamma}(\Omega). \tag{4.5}$$

When the social utility function γ is nondecreasing and submodular, we have $c \in [0, 1]$, which implies that $\bar{\gamma}(S) \geq \bar{\gamma}(\Omega)/2$.

4.3 Nash Equilibria Based on User Groups

4.3.1 Social-Aware Nash Equilibria

In this section, we first introduce the social group utility maximization system and the social-aware Nash equilibrium defined in [2]. Then, we show that the results of [1] are directly applicable to bounding the performance of any social-aware Nash equilibrium.

In [2], each user belongs to a group and aims to maximize its social group utility instead of its private utility. Each group is formed based on social ties between users and may reflect friendship, kinship, college relationship, etc. The social group utility for user i (a mapping from $\mathcal{X}$ to real numbers) is defined as

$$\eta_i = \alpha_i + \sum_{m \in \mathcal{N}_i^s} \omega_{im} \alpha_m$$

where α_i's are private utilities, $\mathcal{N}_i^s$ is the set of all users having a social tie with user i, and w_{im}'s are weight parameters that reflect the strengths of social ties between user i and the users in $\mathcal{N}_i^s$, and $w_{im} \in [0, 1]$. Correspondingly, the expected group utility $\bar{\eta}_i$ for user i, mapping from sequences in $\mathcal{S}$ to real numbers, is the expected value of η_i.

Definition 4.3.1. *[2] A strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$ is a social-aware Nash equilibrium if no user can improve its group utility by unilaterally changing its strategy, i.e., for any group i,*

$$\bar{\eta}_i(S) \geq \bar{\eta}_i((S_{-i}, s_i')), \quad \forall s_i' \in \mathcal{S}_i. \tag{4.6}$$

By comparing the definition of a Nash equilibrium and a social-aware Nash equilibrium, we see that the only difference between them is that one is defined based on expected private utility functions and the other based on expected group utility functions. But because in [2], each user has its own group utility function, and therefore its own expected group utility function, then the results of [1] (in particular Theorem 1 and Theorem 2) directly apply to bound the performance of the social-aware Nash equilibrium of [2]. We prove in Theorem 3 and Theorem 4 that this is in fact the case, if the social group utility system $(\gamma, \{\eta_i\}_{i=1}^N)$ is valid. A social group utility system $(\gamma, \{\eta_i\}_{i=1}^N)$ is valid if it satisfies the following assumptions, which are counterparts of Assumption 4.2.1 and Assumption 4.2.2 with expected group utilities standing in for expected private utilities.

Assumption 4.3.1. *The group utility of user i $(1 \leq i \leq N)$ is at least as large as the loss in the social utility resulting from user i dropping out of the game. That is, the system $(\gamma, \{\eta_i\}_{i=1}^N)$ has the property that for any strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$,*

$$\bar{\eta}_i(S) \geq \bar{\gamma}_{s_i}(S_{-i}), \quad \forall 1 \leq i \leq N. \tag{4.7}$$

Assumption 4.3.2. *The sum of the group utilities of the system is not larger than the social utility, i.e., for any strategy sequence $S = (s_1, \ldots, s_N) \in \mathcal{S}$,*

$$\sum_{i=1}^{N} \bar{\eta}_i(S) \leq \bar{\gamma}(S). \tag{4.8}$$

Given $X \in \mathcal{X}$, if for any $1 \leq i \leq N$, the inequalities $\eta_i(X) \geq \gamma_{x_i}(X_{-i})$ and $\sum_{i=1}^{N} \eta_i(X) \leq \gamma(X)$ hold, then the inequalities (4.7) and (4.8) hold.

Remark 4.3.1. *Comparing Definitions 4.2.1 and 4.3.1, we have that the only difference between a Nash equilibrium and a social-aware Nash equilibrium is that the former is defined in terms of $\bar{\alpha}_i$, and the latter is defined in terms of $\bar{\eta}_i$. So if we take $\bar{\eta}_i$ to play the role of $\bar{\alpha}_i$, then satisfying Assumptions 4.3.1 and 4.3.2 means that the utility system satisfies Assumptions 4.2.1 and 4.2.2. Based on the results of Theorems 4.2.1 and 4.2.2, we have the following Theorems 4.3.1 and 4.3.2.*

Theorem 4.3.1. *For a valid utility system $(\gamma, \{\eta_i\}_{i=1}^N)$, if the social utility function γ is submodular, then for any social-aware Nash equilibrium $S \in \mathcal{S}$ we have*

$$\bar{\gamma}(S) \geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + \sum_{i=1}^{N} \bar{\gamma}_{s_i}(S_{-i} \cup \Omega)\right). \tag{4.9}$$

Theorem 4.3.2. *For a valid utility system $(\gamma, \{\eta_i\}_{i=1}^N)$, if the social utility function γ is nondecreasing and submodular, then for any Nash equilibrium $S \in \mathcal{S}$ we have*

$$\bar{\gamma}(S) \geq \frac{1}{1+c}\bar{\gamma}(\Omega). \tag{4.10}$$

4.3.2 Group Nash Equilibria

In this section we consider a different type of social group utility maximization system in which the set of all users are divided into disjoint groups, and the users in the same group choose their strategies by maximizing their group utility function jointly.

Assume that the set of users $\mathcal{N} = \{1, \ldots, N\}$ is divided into l disjoint groups, in which group i ($1 \leq i \leq l$) has users $\{m_i + 1, \ldots, m_i + k_i\}$, where $m_i = \sum_{j=1}^{i-1} k_j$, k_j is the number of users in group j, and $\sum_{j=1}^{l} k_j = N$. Let $s^i = (s_{m_i+1}, \ldots, s_{m_i+k_i})$, where $s_i \in \mathcal{S}_i$ is the strategy for user i. We call s^i the group strategy for group i. It includes the strategies taken by all the users in group i ($1 \leq i \leq l$). We use S^{-i} to denote the sequence of group strategies taken by all groups except for group i. Given S^{-i}, we denote by (S^{-i}, t^i) the group strategy sequence obtained when group i changes its group strategy from s^i to t^i. Similarly, for $X \in \mathcal{X}$, we use x^i and X^{-i} to denote the sequence of actions taken by the users in group i, and the sequence of actions taken by all groups except for group i, respectively. For convenience, we still use η_i and $\bar{\eta}_i$ to denote the group utility function and the expected group utility function for group i.

We define a group Nash equilibrium as follows.

Definition 4.3.2. *A strategy set $S = (s_1, \ldots, s_N)$ is a group Nash equilibrium of a utility system if no group can improve its group utility by unilaterally changing its group strategy, i.e., for any $1 \leq i \leq l$,*

$$\bar{\eta}_i(S) \geq \bar{\eta}_i((S^{-i}, t^i)), \quad \forall t^i = (t_{m_i+1}, \ldots, t_{m_i+k_i}),$$

where $t_j \in \mathcal{S}_j$ for $m_i + 1 \leq j \leq m_i + k_i$.

We say that the utility system $(\gamma, \{\eta_i\}_{i=1}^{l})$ is valid if it satisfies the following two assumptions.

Assumption 4.3.3. *The group utility of group i is at least as large as the loss in the social utility resulting from all the users in group i dropping out of the game. That is, the system $(\gamma, \{\eta_i\}_{i=1}^{l})$ has the property that for any strategy sequence $S = (s^1, \ldots, s^l) \in \mathcal{S}$,*

$$\bar{\eta}_i(S) \geq \bar{\gamma}_{s^i}(S^{-i}), \quad \forall 1 \leq i \leq l. \tag{4.11}$$

Assumption 4.3.4. *The sum of the group utilities of the system is not larger than the social utility, i.e., for any strategy sequence $S = (s^1, \ldots, s^l) \in \mathcal{S}$,*

$$\sum_{i=1}^{l} \bar{\eta}_i(S) \leq \bar{\gamma}(S). \tag{4.12}$$

Given $X \in \mathcal{X}$, if for any $1 \leq i \leq l$, the inequalities $\eta_i(X) \geq \gamma_{x^i}(X^{-i})$ and $\sum_{i=1}^{l} \eta_i(X) \leq \gamma(X)$ hold, then the inequalities (4.11) and (4.12) hold. We now present our results on the performance of a group Nash equilibrium relative to the optimal social strategy Ω. Although the overall flow of the proof for deriving performance bound (without curvature) for the group Nash equilibria is similar to that of the proof from [1], we still include it here because it will help us derive performance bounds involving curvature later on.

Lemma 4.3.3. *Assume that the social utility function γ is a submodular set function. Then for any strategy set $S \in \mathcal{S}$,*

$$\bar{\gamma}(\Omega) \leq \bar{\gamma}(S) + \sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{\sigma^i}(S^{-i}) - \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(S^{(i-1)} \cup \Omega), \tag{4.13}$$

where $S^{(i)} = s^1 \oplus s^2 \oplus \cdots \oplus s^i$ is the sequence of the group strategies taken by the first i groups.

Proof. Write $\Omega = \sigma^1 \oplus \cdots \oplus \sigma^l$ and $S = s^1 \oplus \cdots \oplus s^l$, where $\sigma^i = (\sigma_{m_i+1}, \ldots, \sigma_{m_i+k_i})$, $s^i = (s_{m_i+1}, \ldots, s_{m_i+k_i})$, and $\sigma_j, s_j \in \mathcal{S}_j$ for $m_i + 1 \leq j \leq m_i + k_i$.

By Propositions 1 and 2 in [50], we have that

$$\begin{aligned}
\bar{\gamma}(\Omega \cup S) &\leq \bar{\gamma}(S) + \sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{\sigma^i}(S) \\
&\leq \bar{\gamma}(S) + \sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{\sigma^i}(S^{-i})
\end{aligned}$$

and

$$\bar{\gamma}(\Omega \cup S) = \bar{\gamma}(\Omega) + \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(S^{(i-1)} \cup \Omega).$$

Combining the two inequalities above, we have (4.13). □

Theorem 4.3.4. *For a valid utility system* $(\gamma, \{\eta_i\}_{i=1}^N)$*, if the social utility function* γ *is submodular, then any group Nash equilibrium* $S = (s_1, \ldots, s_N) \in \mathcal{S}$ *satisfies*

$$\bar{\gamma}(S) \geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + \sum_{i=1}^{l} \bar{\gamma}_{s^i}(\Omega \cup S^{-i})\right). \tag{4.14}$$

Proof. By Lemma 4.3.3, we have

$$\bar{\gamma}(\Omega) \leq \bar{\gamma}(S) + \sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{\sigma^i}(S^{-i}) - \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(S^{(i-1)} \cup \Omega).$$

By the definition of a group Nash equilibrium, we have

$$\sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{\sigma^i}(S^{-i}) \leq \sum_{i:\sigma^i \subseteq \Omega \setminus S} \bar{\gamma}_{s^i}(S^{-i}) \leq \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(S^{-i}).$$

By Assumptions 4.3.3 and 4.3.4, we have

$$\begin{aligned}
\sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(S^{-i}) &\leq \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\eta}_i(S) \\
&\leq \bar{\gamma}(S) - \sum_{i:s^i \subseteq S \cap \Omega} \bar{\eta}_i(S) \\
&\leq \bar{\gamma}(S) - \sum_{i:s^i \subseteq S \cap \Omega} \bar{\gamma}_{s^i}(S^{-i}).
\end{aligned}$$

Combining the inequalities above and using submodularity results in

$$\begin{aligned}
\bar{\gamma}(\Omega) &\leq 2\bar{\gamma}(S) - \sum_{i:s^i \subseteq S \cap \Omega} \bar{\gamma}_{s^i}(S^{-i}) - \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(\Omega \cup S^{(i-1)}) \\
&\leq 2\bar{\gamma}(S) - \sum_{i:s^i \subseteq S \cap \Omega} \bar{\gamma}_{s^i}(\Omega \cup S^{-i}) - \sum_{i:s^i \subseteq S \setminus \Omega} \bar{\gamma}_{s^i}(\Omega \cup S^{-i}) \\
&\leq 2\bar{\gamma}(S) - \sum_{i=1}^{l} \bar{\gamma}_{s^i}(\Omega \cup S^{-i}),
\end{aligned}$$

which implies that the inequality (4.14) holds. $\square$

Remark 4.3.2. *If the utility function γ is nondecreasing, then the term $\sum_{i=1}^{l} \bar{\gamma}_{s^i}(\Omega \cup S^{-i})$ is nonnegative, so $\bar{\gamma}(S) \geq \frac{1}{2}\bar{\gamma}(\Omega)$, which means that the social value of any group Nash equilibrium is at least half of the optimal social utility value.*

To better characterize the relation of the social utility value of any group Nash equilibrium and that of the optimal solution Ω, we define the group curvature c_{k_i} of the social utility function for group i as

$$c_{k_i} = \max_{S \in \mathcal{S}, \bar{\gamma}_{s^i}(\emptyset) \neq 0} \left\{ 1 - \frac{\bar{\gamma}_{s^i}(\Omega \cup S^{-i})}{\bar{\gamma}_{s^i}(\emptyset)} \right\}.$$

Lemma 4.3.5. *Assume tha the utility function γ is submodular and nondecreasing. Then we have $c_{k_i} \leq c$ for $1 \leq i \leq l$. Especially, if $\mathcal{X}_1 = \mathcal{X}_2 = \cdots = \mathcal{X}_N$, then we have $c_{k_i} \leq c_{k_j}$ for $k_i \geq k_j$.*

The proof of $c_{k_i} \leq c$ is similar to that of Theorem 3.3 from [53] and the proof of $c_{k_i} \leq c_{k_j}$ for $k_i \geq k_j$ is similar to that of Theorem 3.4 from [53], so we skip it here.

Lemma 4.3.6. *Assume that γ is a submodular set function. Then for any strategy set $S = (s_1, \ldots, s_N) \in \mathcal{S}$, we have*

$$\bar{\gamma}(S) \leq \sum_{i=1}^{l} \bar{\gamma}_{s^i}(\emptyset)$$

where $s^i = (s_{m_i+1}, \ldots, s_{m_i+k_i})$ for $1 \leq i \leq l$.

Proof. By the submodularity of $\bar{\gamma}$, we have

$$\begin{aligned}
\bar{\gamma}(S) &= \bar{\gamma}_{s^1}(\emptyset) + \bar{\gamma}_{s^2}(s^1) + \cdots + \bar{\gamma}_{s^i}(s^1 \oplus \cdots \oplus s^{i-1}) + \bar{\gamma}_{s^l}(s^1 \oplus \cdots \oplus s^{l-1}) \\
&\leq \bar{\gamma}_{s^1}(\emptyset) + \bar{\gamma}_{s^2}(\emptyset) + \cdots + \bar{\gamma}_{s^i}(\emptyset) + \cdots + \bar{\gamma}_{s^l}(\emptyset) \\
&= \sum_{i=1}^{l} \bar{\gamma}_{s^i}(\emptyset).
\end{aligned}$$

□

Theorem 4.3.7. *For a valid utility system $(\gamma, \{\eta_i\}_{i=1}^{l})$, if the social utility function γ is nondecreasing and submodular, then any group Nash equilibrium $S = (s_1, \ldots, s_N) \in \mathcal{S}$ satisfies*

$$\bar{\gamma}(S) \geq \frac{1}{1+\max\limits_{1\leq i\leq l} c_{k_i}}\bar{\gamma}(\Omega).$$

Especially, if $\mathcal{X}_1 = \mathcal{X}_2 = \cdots = \mathcal{X}_N$*, we have*

$$\bar{\gamma}(S) \geq \frac{1}{1+c_{k^*}}\bar{\gamma}(\Omega),$$

where $k^* = \min_{1\leq i\leq l} k_i$*.*

Proof. For any group Nash equilibrium $S \in \mathcal{S}$, write $S = s^1 \oplus \cdots \oplus s^l$, where $s^i = (s_{m_i+1}, \ldots, s_{m_i+k_i})$ for $1 \leq i \leq l$.

By the definition of the curvature c_{k_i} for group i, we have

$$\bar{\gamma}_{s^i}(\Omega \cup S^{-i}) \geq (1 - c_{k_i})\,\bar{\gamma}_{s^i}(\emptyset).$$

Using the inequality above, Lemma 4.3.6, and Theorem 4.3.4, we have

$$\begin{aligned}
\bar{\gamma}(S) &\geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + \sum_{i=1}^{l}\bar{\gamma}_{s^i}(\Omega \cup S^{-i})\right)\\
&\geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + \sum_{i=1}^{l}(1 - c_{k_i})\,\bar{\gamma}_{s^i}(\emptyset)\right)\\
&\geq \frac{1}{2}\left(\bar{\gamma}(\Omega) + (1 - \max_{1\leq i\leq l} c_{k_i})\sum_{i=1}^{l}\bar{\gamma}_{s^i}(\emptyset)\right)\\
&\geq \frac{1}{2}(\bar{\gamma}(\Omega) + (1 - \max_{1\leq i\leq l} c_{k_i})),
\end{aligned}$$

which implies that

$$\bar{\gamma}(S) \geq \frac{1}{1+\max\limits_{1\leq i\leq l} c_{k_i}}\bar{\gamma}(\Omega).$$

When $\mathcal{X}_1 = \mathcal{X}_2 = \cdots = \mathcal{X}_N$, by Lemma 4.3.5, we have that $c_{k_i} \leq c_{k_j}$ for $k_i \geq k_j$. Therefore, we have

$$\bar{\gamma}(S) \geq \frac{1}{1+c_{k^*}}\bar{\gamma}(\Omega),$$

where $k^* = \min_{1 \le i \le l} k_i$. $\square$

Remark 4.3.3. *When the group utility function γ is non-decreasing and submodular, it is easy to check that $c_{k_i} \in [0, 1]$, which implies that $1/(1 + \max_{1 \le i \le l} c_{k_i}) \ge 1/2$.*

Remark 4.3.4. *When the group utility function γ is non-decreasing and submodular, we have $\bar{\gamma}(S) \ge \bar{\gamma}(\Omega)/(1 + \max_{1 \le i \le l} c_{k_i}) \ge \bar{\gamma}(\Omega)/(1 + c)$. This shows that the bound for the case with grouping is tighter than that for the case without grouping. Of course, this is unsurprising, because grouping entails cooperation. Moreover, under the condition that each user has the same action space, the larger the value of k_i, the higher the degree of cooperation, and the tighter the lower bound.*

Remark 4.3.5. *We point out that each group can be viewed as a new user with vector-valued actions, and a 1/2 bound for the performance of group Nash equilibrium follows from the result of Vetta. But our analysis goes further by defining the group curvature c_{k_i} associated with group i with k_i users; in doing so, we obtain a tighter bound, namely $1/(1 + \max_{1 \le i \le l} c_{k_i})$. In the special case where each user has the same action space, then we have that any group Nash equilibrium achieves at least $1/(1 + c_{k^*})$ of the optimal social utility, where k^* is the least number of users among the l groups, and the larger the value of k^*, the tighter the lower bound.*

4.4 Example

In this section, we consider the application of utility-based maximization in database assisted spectrum access, adopted from [2]. We will show that the utility system is valid and the social utility function is submodular. We then apply the performance bounds for Nash, social-aware Nash, and group Nash equilibria.

Consider a set of users $\mathcal{N} = \{1, \ldots, N\}$ and a set of TV channels $\mathcal{M} = \{1, \ldots, M\}$. The users in $\mathcal{N}$ wish to access the TV channels in $\mathcal{M}$, for purposes other than TV transmissions, in a way that does not unnecessarily disrupt the primary use of these channels, which is for TV transmission. Specifically, to protect the primary TV users, each user i sends a spectrum access request message containing its geo-location information to a geo-location database. In response, the database sends back the set of vacant channels $\mathcal{M}_i \in \mathcal{M}$ and the allowable transmission power level P_i. Then each user i chooses a feasible channel a_i from the vacant channel set $\mathcal{M}_i$ for data transmission. When multiple users choose to access the same vacant channel, they might interfere with each other, depending on their relative distance: If the distance between users m and i is d_{mi}, interference occurs only if $d_{mi} \leq \delta$, where δ is a given threshold. The aim is to minimize the total interference which is the sum of interference received by each user.

For a collection of selected channels $A = (a_1, \ldots, a_N) \in \prod_{i=1}^{N} \mathcal{M}_i$, the interference experienced by user i is defined as

$$I_i(A) = \sum_{m \in \mathcal{N}_i^p} P_m d_{mi}^{-\lambda} I_{\{a_i = a_m\}} + \omega_{a_i}^i,$$

where $\mathcal{N}_i^p$ is the set of users that can interfere with user i, λ is a path-loss factor, $I_{\{\cdot\}}$ is the indicator function, and $\omega_{a_i}^i$ is the noise including the interchannel interference in channel a_i resulting from primary TV users using other channels. The private utility function α_i of user i is then defined as

$$\alpha_i(A) = -I_i(A) = -\sum_{m \in \mathcal{N}_i^p} P_m d_{mi}^{-\lambda} I_{\{a_i = a_m\}} - \omega_{a_i}^i.$$

This private utility reflects the fact that each user desires to minimize its experienced interference. The social group utility of each user i is defined as

$$\eta_i(A) = \alpha_i(A) + \sum_{m \in \mathcal{N}_i^s} w_{im} \alpha_m(A).$$

Finally, the social utility function is $\gamma(A) = \sum_{i=1}^{N} \alpha_i(A)$.

4.4.1 Nash Equilibria

First we will prove that the utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$ satisfies Assumptions 4.2.1 and 4.2.2, and the social utility function $\gamma(A) = \sum_{i=1}^{N} \alpha_i(A)$ is submodular.

To prove that the system $(\gamma, \{\alpha_i\}_{i=1}^N)$ satisfies Assumption 4.2.1, it suffices to prove that for $1 \le i \le N$,

$$\alpha_i(A) \ge \gamma(A) - \gamma(A_{-i}).$$

By the definition of $\alpha_i(A)$, we have that

$$\gamma(A) = -\sum_{i=1}^{N} \sum_{m \in \mathcal{N}_i^p} P_m d_{mi}^{-\lambda} I_{\{a_i = a_m\}} - \sum_{i=1}^{N} \omega_{a_i}^i.$$

Thus,

$$\begin{aligned}\gamma(A) - \gamma(A_{-i}) &= -\sum_{m \in \mathcal{N}_i^p} P_m d_{mi}^{-\lambda} I_{\{a_i = a_m\}} - \sum_{n: i \in \mathcal{N}_n^p} P_i d_{in}^{-\lambda} I_{\{a_n = a_i\}} - \omega_{a_i}^i \\ &= \alpha_i(A) - \sum_{n: i \in \mathcal{N}_n^p} P_i d_{in}^{-\lambda} I_{\{a_n = a_i\}} \\ &\le \alpha_i(A),\end{aligned}$$

which shows that the utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$ satisfies Assumption 4.2.1. Because $\gamma(A) = \sum_{i=1}^N \alpha_i(A)$, the utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$ also satisfies Assumption 4.2.2.

Let $A_k = (a_1, \ldots, a_k)$ and $A_l = A_k \oplus (a_{k+1}, \ldots, a_l)$ $(l < N)$. To prove that $\gamma(A) = \sum_{i=1}^N \alpha_i(A)$ is submodular, it suffices to prove that for any $a_j \in \mathcal{M}_j$ $(l+1 \le j \le N)$,

$$\gamma_{a_j}(A_k) \ge \gamma_{a_j}(A_l).$$

By definition, we have

$$\begin{aligned}\gamma_{a_j}(A_k) &= \gamma(A_k \oplus a_j) - \gamma(A_k) \\ &= -\sum_{m \in \mathcal{N}_j^p, 1 \le m \le k} P_m d_{mj}^{-\lambda} I_{\{a_j = a_m\}} - \sum_{n: j \in \mathcal{N}_n^p, 1 \le n \le k} P_j d_{jn}^{-\lambda} I_{\{a_n = a_j\}} - \omega_{a_j}^j\end{aligned}$$

and

$$\gamma_{a_j}(A_l) = \gamma(A_l \oplus a_j) - \gamma(A_l)$$
$$= - \sum_{m \in \mathcal{N}_j^p, 1 \le m \le l} P_m d_{mj}^{-\lambda} I_{\{a_j = a_m\}} - \sum_{n: j \in \mathcal{N}_n^p, 1 \le n \le l} P_j d_{jn}^{-\lambda} I_{\{a_n = a_j\}} - \omega_{a_j}^j,$$

which implies that

$$\gamma_{a_j}(A_k) \ge \gamma_{a_j}(A_l).$$

We have now established that the utility system $(\gamma, \{\alpha_i\}_{i=1}^N)$ is valid, and the social utility function $\gamma(A) = \sum_{i=1}^N \alpha_i(A)$ is submodular. This implies that the performance bound in Theorem 4.2.1 holds.

4.4.2 Social-Aware Nash Equilibria

Let

$$p = \min_{1 \le j \le N} \{1 + \sum_{i: j \in \mathcal{N}_i^s} w_{ij}\}$$

Because maximizing $\sum_{i=1}^N \alpha_i(A)$ (with respect to $A \in \mathcal{M}$) is equivalent to maximizing $p \sum_{i=1}^N \alpha_i(A)$, for convenience, we set $\gamma(A) = p \sum_{i=1}^N \alpha_i(A)$ when considering the utility system $(\gamma, \{\eta_i\}_{i=1}^N)$.

Now prove that the system satisfies Assumption 4.3.2.

$$\sum_{i=1}^N \eta_i(A) = \sum_{i=1}^N \alpha_i(A) + \sum_{i=1}^N \sum_{n: n \in \mathcal{N}_i^s} \omega_{in} \alpha_n(A)$$
$$= \sum_{j=1}^N (1 + \sum_{i: j \in \mathcal{N}_i^s} w_{ij}) \alpha_j(A)$$
$$\le p \sum_{i=1}^N \alpha_i(A).$$

This implies that the utility system $(\gamma, \{\eta_i\}_{i=1}^N)$ satisfies Assumption 4.3.2.

We now prove that the utility system $(\gamma, \{\eta_i\}_{i=1}^N)$ satisfies Assumption 4.3.1. By the definition of $\gamma(A)$ and $\eta_i(A)$, we have

$$
\begin{aligned}
\gamma(A) - \gamma(A_{-i}) &= p\left(-\sum_{m \in \mathcal{N}_i^p} P_m d_{mi}^{-\lambda} I_{\{a_i = a_m\}} - \sum_{n: i \in \mathcal{N}_n^p} P_i d_{in}^{-\lambda} I_{\{a_n = a_i\}} - \omega_{a_i}^i\right) \\
&= p\left(\alpha_i(A) - \sum_{n: i \in \mathcal{N}_n^p} P_i d_{in}^{-\lambda} I_{\{a_n = a_i\}}\right) \\
&= \alpha_i(A) + \min_{1 \le j \le N}\{\sum_{i: j \in \mathcal{N}_i^s} w_{ij}\}\alpha_i(A) - p \sum_{n: i \in \mathcal{N}_n^p} P_i d_{in}^{-\lambda} I_{\{a_n = a_i\}}.
\end{aligned}
$$

and

$$
\eta_i(A) = \alpha_i(A) + \sum_{n: n \in \mathcal{N}_i^s} w_{in}\alpha_n(A).
$$

For convenience, we consider the case when the transmission power of all the users are the same (i.e., $P_m = P_n = P$ for any users m and n). By Theorem 1 from [2], we have that the social tie between any two users is symmetric (i.e., $w_{nm} = w_{mn}$). Then we can write p and $p(\gamma(A) - \gamma(A_{-i}))$ as follows.

$$
p = \min_{1 \le i \le N}\{1 + \sum_{m \in \mathcal{N}_i^s} w_{im}\}
$$

and

$$
\begin{aligned}
p(\gamma(A) - \gamma(A_{-i})) &= p(\alpha_i(A) - \sum_{m \in \mathcal{N}_i^p} P d_{mi}^{-\lambda} I_{\{a_i = a_m\}}) \\
&= \alpha_i(A) + (\min_{1 \le i \le N} \sum_{m \in \mathcal{N}_i^s} w_{im})\alpha_i(A) + (-p \sum_{m \in \mathcal{N}_i^p} P d_{mi}^{-\lambda} I_{\{a_i = a_m\}}).
\end{aligned}
$$

So only if

$$
\sum_{n: n \in \mathcal{N}_i^s} w_{in}\alpha_n(A) \ge (\min_{1 \le i \le N} \sum_{m \in \mathcal{N}_i^s} w_{im})\alpha_i(A) - p \sum_{m \in \mathcal{N}_i^p} P d_{mi}^{-\lambda} I_{\{a_i = a_m\}} \tag{4.15}
$$

holds, we have that Assumption 4.3.1 holds.

Finally, we have that $\gamma(A) = p\sum_{i=1}^{N}\alpha_i(A)$ is submodular because we proved that $\sum_{i=1}^{N}\alpha_i(A)$ is submodular in Subsection A. So we have now established that if the inequality (4.15) holds, then the utility system $(\gamma, \{\eta_i\}_{i=1}^{N})$ is valid and the social utility function $\gamma(A) = p\sum_{i=1}^{N}\alpha_i(A)$ is submodular. This implies that the performance bound for a social-aware Nash equilibrium in Theorem 4.3.1 holds.

4.4.3 Group Nash Equilibria

We now partition the set of users $\mathcal{N} = \{1, \ldots, N\}$ into l disjoint groups and write, as before, $\sum_{i=1}^{l} k_i = N$ and $m_i = \sum_{j=1}^{i-1} k_j$. Group i comprises the users $\{m_i + 1, \ldots, m_i + k_i\}$, and the group utility function is $\eta_i(A) = \sum_{j=1}^{k_i}\alpha_{m_i+j}(A)$. Finally, the social utility is given by $\gamma(A) = \sum_{i=1}^{N}\alpha_i(A)$.

We now show that the utility system $(\gamma, \{\eta_i\}_{i=1}^{N})$ satisfies Assumption 4.3.3. Let $A = a^1 \oplus \cdots \oplus a^l \in \mathcal{M}$. Then for $1 \le i \le l$,

$$\begin{aligned}
\gamma(A) - \gamma(A^{-i}) &= -\sum_{j=m_i+1}^{m_i+k_i}\sum_{n\in\mathcal{N}_j^p} P_n d_{nj}^{-\lambda} I_{\{a_j=a_n\}} - \sum_{j=m_i+1}^{m_i+k_i}\sum_{n:j\in\mathcal{N}_n^p} P_j d_{jn}^{-\lambda} I_{\{a_n=a_j\}} - \sum_{j=m_i+1}^{m_i+k_i}\omega_{a_j}^{j} \\
&= \eta_i(A) - \sum_{j=m_i+1}^{m_i+k_i}\sum_{n:j\in\mathcal{N}_n^p} P_j d_{jn}^{-\lambda} I_{\{a_n=a_j\}} \\
&\le \eta_i(A),
\end{aligned}$$

which implies that the utility system $(\gamma, \{\eta_i\}_{i=1}^{N})$ satisfies Assumption 4.3.3.

Because $\sum_{i=1}^{l}\eta_i(A) = \sum_{i=1}^{N}\alpha_i(A) = \gamma(A)$, we have that the utility system $(\gamma, \{\eta_i\}_{i=1}^{N})$ also satisfies Assumption 4.3.4. Moreover, we have proved that the social utility $\gamma(A) = \sum_{i=1}^{N}\alpha_i(A)$ is submodular in Subsection A.

We have thus established that the utility system $(\gamma, \{\eta_i\}_{i=1}^{N})$ is valid and the social utility function $\gamma(A) = \sum_{i=1}^{N}\alpha_i(A)$ is submodular. This shows that the performance bound for a group Nash equilibrium in Theorem 4.3.4 holds.

Remark 4.4.1. *The performance bounds we derive here for Nash equilibria, social-aware Nash equilibria, and group Nash equilibria are worst-case performance bounds. The fact that the social-aware group Nash equilibrium derived by [2] achieves 85% of the optimal social utility is consistent with our bound.*

Chapter 5

Performance of Greedy Strategy in String Optimization

In this chapter, we consider an optimization problem where the decision variable is a string of bounded length. For some time there has been an interest in bounding the performance of the greedy strategy for this problem. Here, we provide weakened sufficient conditions for the greedy strategy to be bounded by a factor of $(1-(1-1/K)^K)$, where K is the optimization horizon length. Specifically, in Section 5.1, we introduce the string optimization problem and our motivation. In Section 5.2, we introduce some definitions and review some previous results on performance bounds for the greedy strategy in the string optimization problem. In Section 5.3, we first introduce the notions of K-submodularity and K-GO-concavity, which are sufficient conditions for the bound $(1-(1-1/K)^K)$ to hold. Then we introduce a new notion of curvature $\eta \in (0,1]$ and prove an even tighter bound with the factor $(1/\eta)(1-e^{-\eta})$. In Section 5.4, we illustrate the strength of our results by considering two example applications. We show that our results provide weaker conditions on parameter values in these applications than in previous results. The results in this chapter were published in [57].

5.1 Problem Formulation

In a great number of problems in engineering and applied science, we are faced with optimally choosing a string (finite sequence) of actions over a finite horizon to maximize an objective function. The problem arises in sequential decision making in engineering, economics, management science, and medicine. To formulate the problem precisely, let $\mathbb{A}$ be a set of possible actions. At each stage i, we choose an action a_i from $\mathbb{A}$. Let $A=(a_1,a_2,\ldots,a_k)$ denote a string of actions taken over k consecutive stages, where $a_i \in \mathbb{A}$ for $i=1,2,\ldots,k$. Let $\mathbb{A}^*$ denote the set of all possible strings of actions (of arbitrary length, including the empty string $\varnothing$). Finally, let $f:\mathbb{A}^* \to \mathbb{R}$

be an objective function, where $\mathbb{R}$ denotes the real numbers. Our goal is to find a string $M \in \mathbb{A}^*$, with a length $|M|$ not larger than K (prespecified), to maximize the objective function:

$$\begin{aligned} &\text{maximize } f(M) \\ &\text{subject to } M \in \mathbb{A}^*,\ |M| \leq K. \end{aligned} \tag{5.1}$$

The solution to (5.1), which we call the optimal strategy, is hard to compute in general. One approach is to use dynamic programming via Bellman's principle (see, e.g., [47] and [48]). However, the computational complexity of this approach grows exponentially with the size of $\mathbb{A}$ and the horizon length K. On the other hand, the greedy strategy, though suboptimal in general, is easy to compute because at each stage, we only have to find an action to maximize the step-wise gain in the objective function. But how does the greedy strategy compare with the optimal strategy in terms of the objective function?

The above question has attracted widespread interest, with some key results in the context of string-submodularity (see, e.g., [3,4,63]). These papers extend the celebrated results of Nemhauser et al. [5,6], and some further extensions of them (see, e.g., [34,35,64,65]), on bounding the performance of greedy strategies in maximizing submodular functions over sets, to problem (5.1) that involves maximizing an objective function over strings. In particular, Streeter and Golovin [3] show that if, in (5.1), the objective function f is prefix and postfix monotone and has the diminishing-return property, then the greedy strategy achieves at least a $(1-e^{-1})$-approximation of the optimal strategy. Zhang *et al.* [4] consider a weaker notion of the postfix monotoneity and provide sufficient conditions for the greedy strategy to achieve a factor of at least $(1-(1-1/K)^K)$, where K is the optimization horizon length, of the optimal objective value. They also introduce several notions of curvature, with which the performance bound for the greedy strategy can be further sharpened.

But all the sufficient conditions obtained so far involve strings of length greater than K, even though (5.1) involves only strings up to length K. This motivates a weakening of these sufficient conditions to involve only strings of length at most K, but still preserving the bounds here.

5.2 Review of Related Work

In this section, we first introduce some definitions related to strings and curvature. We then review the main results from [4]. Specifically, the results there provide sufficient conditions on the objective function f in (5.1) such that the greedy strategy achieves a $(1-(1-1/K)^K)$-bound.

5.2.1 Strings and Curvature

For a given string $A = (a_1, a_2, \ldots, a_k)$, we define its length as k, denoted $|A| = k$. If $M = (a_1^m, a_2^m, \ldots, a_{k_1}^m)$ and $N = (a_1^n, a_2^n, \ldots, a_{k_2}^n)$ are two strings in $\mathbb{A}^*$, we write $M = N$ if $|M| = |N|$ and $a_i^m = a_i^n$ for each $i = 1, 2, \ldots, |M|$. Moreover, we define string concatenation as $M \oplus N = (a_1^m, a_2^m, \ldots, a_{k_1}^m, a_1^n, a_2^n, \ldots, a_{k_2}^n)$. If M and N are two strings in $\mathbb{A}^*$, we write $M \preceq N$ if we have $N = M \oplus L$ for some $L \in \mathbb{A}^*$. In this case, we also say that M is a prefix of N.

A function from strings to real numbers, $f : \mathbb{A}^* \to \mathbb{R}$, is string submodular if

i. f has the prefix-monotone property: $\forall M, N \in \mathbb{A}^*$, $f(M \oplus N) \geq f(M)$.

ii. f has the diminishing-return property: $\forall M \preceq N \in \mathbb{A}^*, \forall a \in \mathbb{A}$, $f(M \oplus (a)) - f(M) \geq f(N \oplus (a)) - f(N)$.

A function from strings to real numbers, $f : \mathbb{A}^* \to \mathbb{R}$, is postfix monotone if

$$\forall M, N \in \mathbb{A}^*, f(M \oplus N) \geq f(N).$$

The total backward curvature of f is defined as

$$\sigma = \max_{a \in A, M \in \mathbb{A}^*} \left\{ \frac{(f((a)) - f(\emptyset)) - (f((a) \oplus M) - f(M)))}{f((a)) - f(\emptyset)} \right\}.$$

5.2.2 Bounds for the Greedy Strategy

We now define optimal and greedy strategies for problem (5.1) and some related notation.

(1) Optimal strategy: Any solution to (5.1) is called an optimal strategy. If f is prefix monotone, then there exists an optimal strategy with length K, denoted $O_K = (o_1, \ldots, o_K)$. Let $O_i = (o_1, \ldots, o_i)$ for $i = 1, \ldots, K$.

(2) Greedy strategy: A string $G_k = (g_1, g_2, \ldots, g_k)$ is called a greedy strategy if $\forall i = 1, 2, \ldots, k$,

$$g_i \in \operatorname*{argmax}_{g \in \mathbb{A}} f((g_1, g_2, \ldots, g_{i-1}, g)).$$

Let $G_i = (g_1, \ldots, g_i)$ for $i = 1, \ldots, K$.

The following two theorems summarize the performance bounds in [4].

Theorem 5.2.1. *If f is string submodular and $f(G_i \oplus O_K) \geq f(O_K)$ holds for all $i = 1, \ldots, K-1$, then any greedy strategy G_K satisfies*

$$f(G_K) \geq \left(1 - \left(1 - \frac{1}{K}\right)^K\right) f(O_K) > (1 - e^{-1}) f(O_K).$$

Theorem 5.2.2. *If f is string submodular and postfix monotone, then any greedy strategy G_K satisfies*

$$\begin{aligned} f(G_K) &\geq \frac{1}{\sigma}\left(1 - \left(1 - \frac{\sigma}{K}\right)^K\right) f(O_K) \\ &> \frac{1}{\sigma}(1 - e^{-\sigma}) f(O_K) \\ &> (1 - e^{-1}) f(O_K). \end{aligned}$$

Under additional assumptions on the curvature σ of f, [4] provide even tighter bounds. Notice that the sufficient conditions above involve strings of length greater than K, even though the problem (5.1) involves only strings up to length K. This motivates a weakening of these sufficient conditions to involve only strings of length at most K, but still preserving the bounds here. In the next section, we present our main results along these lines. In Section 5.4, we show that these weakened sufficient conditions also lead to weaker requirements than in [4] for two application examples.

5.3 Main Results

Before stating our main results, we first introduce some definitions on $f : \mathbb{A}^* \to \mathbb{R}$.

i f is K-monotone if $\forall M, N \in \mathbb{A}^*$, and $|M| + |N| \leq K$, $f(M \oplus N) \geq f(M)$.

ii. f is K-diminishing if $\forall M \preceq N \in \mathbb{A}^*$ and $|N| \leq K-1$, $\forall a \in \mathbb{A}$, $f(M \oplus (a)) - f(M) \geq f(N \oplus (a)) - f(N)$.

iii. f is K-submodular if it is both K-monotone and K-diminishing.

iv. Let $G_i = (g_1, \ldots, g_i)$ (as before) and $\bar{O}_{K-i} = (o_{i+1}, \ldots, o_K)$ for $i = 1, \ldots, K$. Then, f is K-GO-concave if for $1 \leq i \leq K-1$,

$$f(G_i \oplus \bar{O}_{K-i}) \geq \frac{i}{K} f(G_i) + \left(1 - \frac{i}{K}\right) f(O_K).$$

Notice that these definitions involve only strings of length at most K. Moreover, it is clear that if f is string submodular, prefix monotone, and has the diminishing-return property, then f is string K-submodular, K-monotone, and K-diminishing. Under these weaker conditions, we show that the previous bounds on the greedy strategy still hold.

Theorem 5.3.1. *If f is K-submodular and K-GO-concave, then*

$$f(G_K) \geq \left(1 - \left(1 - \frac{1}{K}\right)^K\right) f(O_K) > (1 - e^{-1}) f(O_K).$$

Proof. Because f is K-diminishing, we have that for $1 \leq i \leq K$,

$$f(o_i) \geq f(O_i) - f(O_{i-1}).$$

By definition of the greedy strategy, for $1 \leq i \leq K$,

$$f(G_1) \geq f(o_i) \geq f(O_i) - f(O_{i-1}).$$

Summing the inequality above over i from 1 to K produces

$$\begin{aligned} \sum_{i=1}^{K} f(G_1) &\geq \sum_{i=1}^{K} (f(O_i) - f(O_{i-1})) \\ \Rightarrow \quad K f(G_1) &\geq f(O_K) \\ \Rightarrow \quad f(G_1) &\geq \frac{1}{K} f(O_K). \end{aligned}$$

For $1 \leq i \leq K-1$, because f is K-diminishing, we have

$$f(G_i \oplus o_K) - f(G_i) \geq f(G_i \oplus \bar{O}_{K-j}) - f(G_i \oplus \bar{O}_{K-(j+1)})$$

for $i \leq j \leq K-1$. Summing the inequality above over j, we have

$$\begin{aligned} &(K-i)(f(G_i \oplus o_K) - f(G_i)) \\ &\qquad \geq \sum_{j=i}^{K-1} f(G_i \oplus \bar{O}_{K-j}) - f(G_i \oplus \bar{O}_{K-(j+1)}) \\ &\qquad = f(G_i \oplus \bar{O}_{K-i}) - f(G_i), \end{aligned}$$

which implies that

$$f(G_i \oplus o_K) - f(G_i) \geq \frac{1}{K-i}(f(G_i \oplus \bar{O}_{K-i}) - f(G_i)). \tag{5.2}$$

By K-GO-concavity, for $1 \leq i \leq K-1$ we have

$$\begin{aligned} f(G_i \oplus o_K) - f(G_i) &\geq \frac{1}{K-i}(f(G_i \oplus \bar{O}_{K-i}) - f(G_i)) \\ &\geq \frac{1}{K-i}\left(\frac{K-i}{K}f(O_K) + \frac{i}{K}f(G_i) - f(G_i)\right) \\ &= \frac{1}{K}(f(O_K) - f(G_i)). \end{aligned}$$

Again by definition of the greedy strategy, we have for $1 \leq i \leq K-1$,

$$\begin{aligned} f(G_{i+1}) - f(G_i) &\geq f(G_i \oplus o_K) - f(G_i) \\ &\geq \frac{1}{K}(f(O_K) - f(G_i)) \end{aligned}$$

from which we get

$$f(G_{i+1}) \geq \frac{1}{K}f(O_K) + \left(1 - \frac{1}{K}\right) f(G_i).$$

Therefore,

$$\begin{aligned} f(G_K) &\geq \frac{1}{K}f(O_K) + \left(1 - \frac{1}{K}\right) f(G_{K-1}) \\ &\vdots \\ &\geq \frac{1}{K}f(O_K)\sum_{i=0}^{K-1}\left(1 - \frac{1}{K}\right)^i \\ &= \left(1 - \left(1 - \frac{1}{K}\right)^K\right) f(O_K). \end{aligned}$$

Because $1 - \left(1 - \frac{1}{K}\right)^K \searrow 1 - e^{-1}$ as $K \to \infty$, we also have

$$f(G_K) \geq \left(1 - \left(1 - \frac{1}{K}\right)^K\right) f(O_K) > (1 - e^{-1})f(O_K).$$

$\square$

Next, we introduce a new notion of curvature η as follows:

$$\eta = \max_{1 \le i \le K-1} \left\{ \frac{Kf(G_i) - (Kf(G_i \oplus \bar{O}_{K-i}) - (K-i)f(O_K))}{(K-i)f(G_i)} \right\}.$$

If f is K-GO-concave, then for $1 \le i \le K-1$ we have

$$\begin{aligned} Kf(G_i) - (Kf(G_i \oplus \bar{O}_{K-i}) - (K-i)f(O_K)) &\le Kf(G_i) - if(G_i) \\ &= (K-i)f(G_i), \end{aligned}$$

which implies that $\eta \le 1$. The following theorem gives a bound related to the curvature η.

Theorem 5.3.2. *If f is K-submodular and K-GO-concave, then*

$$\begin{aligned} f(G_K) &\ge \frac{1}{\eta}\left(1 - \left(1 - \frac{\eta}{K}\right)^K\right) f(O_K) \\ &> \frac{1}{\eta}(1 - e^{-\eta}) f(O_K). \end{aligned}$$

Proof. By definition of the curvature η, we have

$$f(G_i \oplus \bar{O}_{K-i}) - f(G_i) \ge \frac{K-i}{K}(f(O_K) - \eta f(G_i)).$$

By definition of the greedy strategy and inequality (5.2), we have

$$\begin{aligned} f(G_{i+1}) - f(G_i) &\ge f(G_i \oplus o_K) - f(G_i) \\ &\ge \frac{1}{K-i} \cdot \frac{K-i}{K}(f(O_K) - \eta f(G_i)) \\ &= \frac{1}{K}(f(O_K) - \eta f(G_i)) \end{aligned}$$

from which we get

$$f(G_{i+1}) \ge \frac{1}{K} f(O_K) + \left(1 - \frac{\eta}{K}\right) f(G_i).$$

Therefore,

$$
\begin{aligned}
f(G_K) &\geq \frac{1}{K} f(O_K) + \left(1 - \frac{\eta}{K}\right) f(G_{K-1}) \\
&\ \ \vdots \\
&\geq \frac{1}{K} f(O_K) \sum_{i=0}^{K-1} \left(1 - \frac{\eta}{K}\right)^i \\
&= \frac{1}{\eta} \left(1 - \left(1 - \frac{\eta}{K}\right)^K\right) f(O_K).
\end{aligned}
$$

Because $\frac{1}{\eta}\left(1 - \left(1 - \frac{\eta}{K}\right)^K\right) \searrow \frac{1}{\eta}(1 - e^{-\eta})$ as $K \to \infty$, we also have

$$
\begin{aligned}
f(G_K) &\geq \frac{1}{\eta} \left(1 - \left(1 - \frac{\eta}{K}\right)^K\right) f(O_K) \\
&> \frac{1}{\eta}(1 - e^{-\eta}) f(O_K).
\end{aligned}
$$

□

Remark 5.3.1. *The term* $\frac{1}{\eta}(1 - e^{-\eta})$ *is decreasing in* $\eta \in (0, 1]$.

Remark 5.3.2. *When* $\eta = 1$, $\frac{1}{\eta}(1 - e^{-\eta}) = 1 - e^{-1}$, *which is the bound in Theorem 5.3.1. Moreover, for* $0 < \eta < 1$, $\frac{1}{\eta}(1 - e^{-\eta}) > 1 - e^{-1}$. *Hence, Theorem 5.3.2 is a generalization of Theorem 5.3.1 and gives a tighter bound.*

Remark 5.3.3. *When* $\eta \to 0$, *we have* $\frac{1}{\eta}(1 - e^{-\eta}) \to 1$, *making the greedy strategy asymptotically optimal.*

5.4 Applications

In this section, we consider two example applications, namely task assignment and adaptive measurement design, to illustrate the strength of our results. In each case, we derive sufficient conditions, on the parameter values of the problem, for the greedy strategy to achieve the $(1 - (1 - 1/K)^K)$ bound. These sufficient conditions are weaker than those we previously reported in [4].

5.4.1 Task Assignment Problem

As our first example application, we consider the task assignment problem that was posed in [3] and was further analyzed in [4]. In this problem, we have n subtasks and a set $\mathbb{A}$ of K agents. At each stage, we get to assign a subtask to an agent, who accomplishes the task with some probability. Let $p_i^j(a)$ denote the probability of accomplishing subtask i at stage j when it is assigned to agent $a \in \mathbb{A}$. Assume that $p_i^j(a) \in [L_i(a), U_i(a)]$, $0 < L_i(a) < U_i(a) < 1$, and that the limits of the interval are independent of the stage in which subtask i is assigned to agent a. Let $X_i(a_1, a_2, \ldots, a_k)$ denote the random variable that describes whether or not subtask i has been accomplished after the sequence of assignments $(a_1, a_2, \ldots, a_k)$ over k steps. Then, $\frac{1}{n}\sum_{i=1}^n X_i(a_1, a_2, \ldots, a_k)$ is the fraction of subtasks accomplished after k steps by employing agents $(a_1, a_2, \ldots, a_k)$ over k steps. The objective function f for this problem is the expected value of this fraction, which can be written as

$$f((a_1, \ldots, a_k)) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \prod_{j=1}^{k}\left(1 - p_i^j(a_j)\right)\right).$$

We wish to derive sufficient conditions on the set of parameters $\{(L(a), U(a)) \,|a \in \mathbb{A}\}$ so that f is K-monotone, K-diminishing, and K-GO-concave.

For simplicity, we consider the case of $n = 1$. But our results can easily be generalized to the case where $n > 1$. For $n = 1$, the objective function f reduces to

$$f((a_1, \ldots, a_k)) = 1 - \prod_{j=1}^{k}(1 - p_1^j(a_j)), \tag{5.3}$$

and from here on we simply use $p^j(a_j)$ in place of $p_1^j(a_j)$.

It is easy to check that f is K-monotone. For f to be K-diminishing, it suffices to have

$$f(M \oplus a) - f(M) \geq f(M \oplus b \oplus a) - f(M \oplus b),$$

for any $a, b \in \mathbb{A}$ and for any $M \in \mathbb{A}^*$ with $|M| \leq K - 2$. Let $M = (a_1, \ldots, a_m)$, then we have

$$p^{m+1}(a) \geq (1 - p^{m+1}(b))p^{m+2}(a).$$

Suppose that $L(a) \leq p^j(a) \leq U(a)$ for all $a \in \mathbb{A}, j = 1, 2, \ldots, K$. Let

$$\hat{U} = \max_{a \in \mathbb{A}} U(a)$$

and

$$\hat{L} = \min_{a \in \mathbb{A}} L(a).$$

Then, we can write

$$p^{m+1}(a) \geq L(a) \geq \hat{L}$$

and

$$(1 - p^{m+1}(b))p^{m+2}(a) \leq (1 - L(b))U(a) \leq (1 - \hat{L})\hat{U}.$$

Thus, a sufficient condition for f to be K-diminishing is

$$\hat{L} \geq (1 - \hat{L})\hat{U}. \tag{5.4}$$

Now, let us rearrange the K-GO-concavity condition as

$$(K - i)(f(O_K) - f(G_i \oplus \bar{O}_{K-i})) \leq i(f(G_i \oplus \bar{O}_{K-i}) - f(G_i)).$$

Replacing for f from (5.3) gives (after simplifying)

$$(K - i) \prod_{j=i+1}^{K} (1 - p^j(o_i)) \left[1 - \frac{\prod_{j=1}^{i}(1 - p^j(o_i))}{\prod_{j=1}^{i}(1 - p^j(g_j))}\right] \leq i \left[1 - \prod_{j=i+1}^{K} (1 - p^j(o_i))\right].$$

Because $f(O_K) \geq f(G_i \oplus \bar{O}_{K-i})$, we have

$$\frac{\prod_{j=1}^{i}(1 - p^j(o_i))}{\prod_{j=1}^{i}(1 - p^j(g_j))} \leq 1.$$

Therefore, to have K-GO-concavity it suffices to have

$$(K-i)\prod_{j=i+1}^{K}(1-p^j(o_i)) \leq i\left[1-\prod_{j=i+1}^{K}(1-p^j(o_i))\right],$$

or equivalently

$$\prod_{j=i+1}^{K}(1-p^j(o_i)) \leq \frac{i}{K}, \tag{5.5}$$

for $1 \leq i \leq K-1$. If we assume that $p^i(o_i) \geq \frac{1}{i}$ for $2 \leq i \leq K$, then it is easy to see that (5.5) holds for $1 \leq i \leq K-1$. Thus, a sufficient condition for K-GO-concavity is

$$\hat{L} \geq \frac{1}{2}. \tag{5.6}$$

If (5.6) holds then (5.4) also holds. Thus, (5.6) is sufficient for the greedy strategy to achieve the $(1-(1-\frac{1}{K})^K)$ bound.

Remark 5.4.1. *The sufficient condition in [4] requires* (5.4) *and*

$$p^1(g_1) \geq 1-c^K, \tag{5.7}$$

where

$$c = \min_{a\in\mathbb{A}} \frac{1-U(a)}{1-L(a)}.$$

When all $p^j(a_j) \geq 1/2$, then (5.6) *and* (5.4) *automatically hold, but* (5.7) *is not necessarily satisfied. In that sense, the K-monotone, K-diminishing, and K-Go concavity conditions are weaker sufficient conditions for achieving the $(1-(1-\frac{1}{K})^K)$ bound than the prefix monotone, diminishing-return, and postfix monotone conditions of [4].*

5.4.2 Adaptive Measurement Problem

As our second example application, we consider the adaptive measurement design problem posed in [30] and [4]. Consider a signal of interest $x \in \mathbb{R}^2$ with normal prior distribution $\mathcal{N}(0, I)$, where I is the 2×2 identity matrix; our analysis easily generalizes to dimensions larger than 2. Let $\mathbb{A} = \{\text{Diag}\left(\sqrt{e}, \sqrt{1-e}\right) \ : \ e \in [0.5, 1]\}$. At each stage i, we make a measurement y_i of the form

$$y_i = A_i x + w_i,$$

where $A_i \in \mathbb{A}$ and w_i is a Gaussian measurement noise vector with mean zero and covariance $\sigma_i^2 I$.

The objective is to choose a string of measurement matrices $\{A_i\}_{i=1}^k$ with $k \leq K$ to maximize the information gain:

$$f((a_1, \ldots, a_k)) = H_0 - H_k.$$

Here $H_0 = \frac{N}{2}\log(2\pi e)$ is the entropy of the prior distribution of x and H_k is the entropy of the posterior distribution of x given $\{y_i\}_{i=1}^k$; that is,

$$H_k = \frac{1}{2}\log \det(P_k) + \frac{N}{2}\log(2\pi e),$$

where

$$P_k = \left(P_{k-1}^{-1} + \frac{1}{\sigma_k^2} A_k^T A_k\right)^{-1}$$

is the posterior covariance of x given $\{y_i\}_{i=1}^k$ [30].

We wish to derive sufficient conditions on the set of parameters $\{\sigma_i^2\}_{i=1}^K$ so that f is K-monotone, K-diminishing, and K-GO-concave. It is easy to see that f is K-monotone by form, and it is K-diminishing if $\{\sigma_i^2\}_{i=1}^K$ is a non-decreasing sequence, that is,

$$\sigma_{i+1}^2 \geq \sigma_i^2, \ \text{ for } i = 1, 2, \ldots, K-1. \tag{5.8}$$

Let $A_i^g = \text{Diag}(\sqrt{e_i}, \sqrt{1-e_i})$ and $A_i^* = \text{Diag}(\sqrt{e_i^*}, \sqrt{1-e_i^*})$ be the greedy and optimal actions at stage i, respectively; that is, $g_i = A_i^g$ and $o_i = A_i^*$. Then, the K-GO-concavity condition for this problem is that for $1 \leq i \leq K-1$, we must have

$$(S_i^* + \bar{S}_{K-i}^*)^{K-i}(c_K - (S_i^* + \bar{S}_{K-i}^*))^{K-i} S_i^i (a_i - S_i)^i \leq (S_i + \bar{S}_{K-i}^*)^K (c_K - (S_i + \bar{S}_{K-i}^*))^K, \tag{5.9}$$

where

$$\begin{aligned}
S_i^* =& 1 + \sum_{j=1}^{i} \frac{1}{\sigma_j^2} e_j^*, \\
\bar{S}_{K-i}^* =& \sum_{j=i+1}^{K} \frac{1}{\sigma_j^2} e_j^*, \\
S_i =& 1 + \sum_{j=1}^{i} \frac{1}{\sigma_j^2} e_j, \\
a_i =& 2 + \sum_{j=1}^{i} \frac{1}{\sigma_j^2}, \\
c_K =& 2 + \sum_{j=1}^{K} \frac{1}{\sigma_j^2}.
\end{aligned}$$

Because $f(O_K) \geq f(G_i \oplus \bar{O}_{K-i})$, we have

$$(S_i^* + \bar{S}_{K-i}^*)(c_K - (S_i^* + \bar{S}_{K-i}^*)) \geq (S_i + \bar{S}_{K-i}^*)(c_K - (S_i + \bar{S}_{K-i}^*)).$$

It is easy to check that

$$S_i(a_i - S_i) \leq (S_i + \bar{S}_{K-i}^*)(c_K - (S_i + \bar{S}_{K-i}^*)).$$

Therefore, we have

$$S_i(a_i - S_i) \leq (S_i^* + \bar{S}_{K-i}^*)(c_K - (S_i^* + \bar{S}_{K-i}^*)). \tag{5.10}$$

Let

$$g(i) = (S_i^* + \bar{S}_{K-i}^*)^{K-i}(c_K - (S_i^* + \bar{S}_{K-i}^*))^{K-i} S_i^i (a_i - S_i)^i.$$

Then,

$$\frac{g(i+1)}{g(i)} = \frac{S_i(a_i - S_i)}{(S_i^* + \bar{S}_{K-i}^*)(c_K - (S_i^* + \bar{S}_{K-i}^*))}.$$

By (5.10), $g(i)$ is non-increasing. Hence, it suffices to have

$$(S_1^* + \bar{S}_{K-1}^*)^{K-1}(c_K - (S_1^* + \bar{S}_{K-1}^*))^{K-1} S_1(a_1 - S_1) \leq \\ (S_1 + \bar{S}_{K-1}^*)^K (c_K - (S_1 + \bar{S}_{K-1}^*))^K \qquad (5.11)$$

in order to get K-GO-concavity.

Let $T_1 = a_1 - S_1$, $T_1^* = a_1 - S_1^*$, and $\bar{T}_{K-1}^* = \sum_{j=2}^{K} \frac{1-e_j^*}{\sigma_j^2}$. Then, we can rewrite (5.11) as

$$(S_1^* + \bar{S}_{K-1}^*)^{K-1}(T_1^* + \bar{T}_{K-1}^*))^{K-1} S_1 T_1 \leq (S_1 + \bar{S}_{K-1}^*)^K (T_1 + \bar{T}_{K-1}^*)^K. \qquad (5.12)$$

If

$$(S_1^* + \bar{S}_{K-1}^*)(T_1^* + \bar{T}_{K-1}^*)) = (S_1 + \bar{S}_{K-1}^*)(T_1 + \bar{T}_{K-1}^*)), \qquad (5.13)$$

that is, to have $f(O_1) = f(G_1)$, then (5.12) always holds, because $S_1 T_1 \leq (S_1 + \bar{S}_{K-1}^*)(T_1 + \bar{T}_{K-1}^*))$.

We now show that (5.13) always holds, given the action set $\mathbb{A}$ considered in this example. In other words, the K-GO-concavity condition is satisfied and this means that (5.8) is a sufficient condition for achieving the $(1 - (1 - \frac{1}{K})^K)$ bound.

By definition of the greedy strategy, we have $f(G_1) \geq f(O_1)$, which means

$$\left(1 + \frac{1}{\sigma_1^2} e_1\right)\left(1 + \frac{1}{\sigma_1^2}(1 - e_1)\right) \geq \left(1 + \frac{1}{\sigma_1^2} e_1^*\right)\left(1 + \frac{1}{\sigma_1^2}(1 - e_1^*)\right).$$

Simplifying the above inequality gives

$$(e_1 - e_1^*)(1 - (e_1 + e_1^*)) \geq 0. \qquad (5.14)$$

Because $f(O_K) \geq f(G_1 \oplus \bar{O}_{k-1})$, we have

$$\left(1+\frac{e_1^*}{\sigma_1^2}+\sum_{j=2}^{K}\frac{e_j^*}{\sigma_j^2}\right)\left(1+\frac{(1-e_1^*)}{\sigma_1^2}+\sum_{j=2}^{K}\frac{(1-e_j^*)}{\sigma_j^2}\right)$$
$$\geq$$
$$\left(1+\frac{e_1}{\sigma_1^2}+\sum_{j=2}^{K}\frac{e_j^*}{\sigma_j^2}\right)\left(1+\frac{(1-e_1)}{\sigma_1^2}+\sum_{j=2}^{K}\frac{(1-e_j^*)}{\sigma_j^2}\right),$$

which implies that

$$(e_1-e_1^*)\left[\sum_{j=2}^{K}\frac{1}{\sigma_j^2}(2e_j^*-1)\right] \geq \frac{1}{\sigma_1^2}(e_1-e_1^*)(1-(e_1+e_1^*)). \tag{5.15}$$

The inequality (5.14) implies that $e_1 \leq e_1^*$. From (5.15) and (5.14), we have that

$$(e_1-e_1^*)\left[\sum_{j=2}^{K}\frac{1}{\sigma_j^2}(2e_j^*-1)\right] \geq 0,$$

which implies that $e_1 = e_1^*$. Since if $e_1 \neq e_1^*$, then $e_1 < e_1^*$, which implies that

$$(e_1-e_1^*)\left[\sum_{j=2}^{K}\frac{1}{\sigma_j^2}(2e_j^*-1)\right] < 0,$$

while

$$\frac{1}{\sigma_1^2}(e_1-e_1^*)\left(1-(e_1+e_1^*)\right) > 0,$$

which contradicts (5.15). Hence, we have $e_1 = e_1^*$, which means $G_1 = O_1$, and the inequality (5.13) holds.

Remark 5.4.2. *The sufficient condition in [4] for achieving the* $(1-(1-\frac{1}{K})^K)$ *bound in this problem requires both* (5.8) *and*

$$\frac{b^{-2}}{a^{-2}-b^{-2}} \geq \frac{(2K-2)^2}{4}(a^{-2}+b^{-2})+1, \tag{5.16}$$

where $[a, b]$ is the interval that contains all the σ_is. Therefore, the condition derived in this paper is a weaker sufficient condition than that obtained in [4].

Chapter 6

A General Framework for Bounding Approximate Dynamic Programming Schemes

In this chapter, we consider a broad family of control strategies called path-dependent action optimization (PDAO), where every control decision is treated as the solution to an optimization problem with a path-dependent objective function. We develop a framework to bound the performance of PDAO schemes. By a bound we mean a guarantee of the form that the performance of a given PDAO scheme relative to the optimal is at least some known factor (typically at least 63%, as we will see soon). The ability to obtain a bound of this kind has enormous implications for artificial-intelligence systems based on PDAO. For example, for the celebrated program AlphaGo [66], we can answer questions such as, "How far from optimal is AlphaGo?", "How much better can AlphaGo get?", and "Is it worth spending much more time and effort to improve AlphaGo?".

Our bounding method is based on the theory of submodular optimization [4]. The basic result from submodular optimization is that in such problems, every greedy scheme can be bounded in the sense outlined above (namely, that it is at least a known factor relative to optimal, typically at least 63%). It turns out that every PDAO scheme is a greedy scheme for some optimization problem. If that optimization problem is equivalent to our problem of interest and is provably submodular (in a certain sense to be made precise later), then we can say with certainty that our PDAO scheme is no worse than something like 63% of optimal.

Our bounding result can be used as a way to check that a PDAO scheme is good—to wit, a PDAO scheme is good if it has the submodular property described above, and hence is guaranteed to be at least a known factor of optimal. Importantly, we can do this check even before we proceed with extensive simulation or testing of the scheme.

Finally, we show how to apply our framework to stochastic optimal control problems (Markov decision processes (MDPs). The family of PDAO schemes of interest here is often called approximate dynamic programming (ADP). Such schemes are based on approximating the second term on the right-hand side of Bellman's optimality principle (the expected value-to-go) by computationally tractable means. Although a wide range of approximate dynamic programming (ADP) methods have been developed [47–49], a general systematic technique to provide performance guarantees for them has remained elusive. Ours is the first systematic approach to deriving performance bounds for general ADP methods in the stochastic setting.

This chapter is organized as follows. In Section 6.1, we review some related previous work and formulate stochastic optimization problems, optimal scheme, PDAO scheme, and greedy policy selection scheme for the stochastic model. We also introduce some terminology and corresponding definitions that will be used in this chapter. In Section 6.2, we provide the framework to derive performance bounds for any greedy policy selection scheme, and prove that any PDAO scheme is also a greedy policy selection scheme, thus performance bounds for any PDAO scheme is obtained. In Section 6.3, we apply our framework to bounding ADP schemes in stochastic optimal control problems.

A portion of results in this chapter were published in [67].

6.1 Preliminaries

In this section, we first review some related previous results, then we formulate a general class of stochastic optimization problems, then define the optimal scheme, PDAO scheme, and the greedy policy-selection scheme for the stochastic model. We also introduce some definitions that will be used in this paper.

6.1.1 Review of Previous Work

Submodularity theory plays an important role in discrete optimization (see, e.g., [3,5,16,22,34, 35,52,68–73]). Under submodularity, the greedy strategy for solving a combinatorial optimization problem provides at least a constant-factor approximation to the optimal strategy. For example, the celebrated result of Nemhauser et al. [6] states that for maximizing a monotone submodular function over a uniform matroid, the objective value of the greedy strategy is no less than a factor $(1 - e^{-1})$ of that of the optimal strategy. The concept of submodularity was extended to functions defined over strings [3, 4, 74], leading to similar bounds on the performance of greedy strategies relative to the optimal strategy in sequential optimization problems, where the objective function depends on the order of actions. In [74], the notion of submodularity for solving stochastic optimization problems was introduced, where the problem is to select a set of actions to maximize an expected reward. Our model generalizes this recent development to path-dependent problems, where the objective function depends on the state trajectory and the order of actions taken.

6.1.2 Problem Formulation

Our aim is to analyze the performance of PDAO schemes as approximately optimal solutions of stochastic optimization problems. But before we formulate the stochastic model, we start with a deterministic model to help motivate our stochastic formulation.

To begin, let $\mathcal{X}$ denote a set of states and $\mathcal{A}$ a set of control actions. Given $x_1 \in \mathcal{X}$ and functions $h : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ and $g : \mathcal{X}^K \times \mathcal{A}^K \to \mathbb{R}_+$, consider the optimization problem

$$\begin{aligned} &\underset{a_1,\ldots,a_K \in \mathcal{A}}{\text{maximize}} \;\; g(x_1, \ldots, x_K; a_1, \ldots, a_K) \\ &\text{s. t.} \;\; x_{k+1} = h(x_k, a_k), \; k = 1, \ldots, K-1. \end{aligned} \tag{6.1}$$

Think of a_k as the control action applied at time k and x_k the state visited at time k. The real number $g(x_1, \ldots, x_K; a_1, \ldots, a_K)$ is the total reward by applying the string of actions a_k at state x_k for $k = 1, \ldots, K$. The function h represents the state-transition law. This model covers a wide variety of optimization problems found in many areas, ranging from engineering to economics. In particular, many adaptive sensing problems have this form (see, e.g., [44]).

We now turn our attention to a stochastic version of problem (6.1), building on the above deterministic case. The key difference is that the state evolves randomly over time in response to actions, whose distribution is specified by the state transition law $x_{k+1} = h(x_k, a_k, \xi_k)$, $k = 1, \ldots, K-1$, where x_1 is a given initial state and $\{\xi_k\}_{k=1}^{K-1}$ is an i.i.d. random sequence. With this modification, we need to change the objective function to $\mathrm{E}[g(x_1, \ldots, x_K; a_1, \ldots, a_K)|x_1]$, involving expectation, where $\mathrm{E}[\cdot|x_1]$ represents conditional expectation given the initial state x_1.

With the specification above, the sequence of states $\{x_k\}_{k=1}^{K}$ has a "Markovian" property in the usual sense. Note that at each time k, the distribution of x_{k+1} depends not only on x_k but also on the control action a_k. Similarly, the total reward function also depends on states and actions. We allow the action at time k to depend on the state x_k. This reduces the optimization problem to one of finding, for each time k, an optimal mapping $\pi_k^* : \mathcal{X} \to \mathcal{A}$, so that the optimal action is given by $a_k = \pi_k^*(x_k)$, corresponding to a state-feedback control law. This mapping is often called a policy (or, sometimes, a Markovian policy).

Define $\pi_k : \mathcal{X} \to \mathcal{A}$ for $k = 1, \ldots, K$, and then treat the string of policies $\pi_1, \ldots, \pi_K$ as the decision variable. For convenience we will also refer to the entire string $(\pi_1, \ldots, \pi_K)$ as simply a policy. The stochastic optimization problem can be formulated in the following form:

$$\begin{aligned} &\underset{\pi_1,\ldots,\pi_K}{\text{maximize}} \quad \mathrm{E}[g(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K))|x_1] \\ &\text{s. t.} \;\; x_{k+1} = h(x_k, \pi_k(x_k), \xi_k), \;\; k = 1, \ldots, K-1. \end{aligned} \tag{6.2}$$

Optimal Scheme: The policy $(\pi_1^*, \ldots, \pi_K^*)$ is optimal if

$$(\pi_1^*, \ldots, \pi_K^*) \in \underset{\pi_1,\ldots,\pi_K}{\operatorname{argmax}} \mathrm{E}[g(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K)|x_1],$$

where $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$ for $1 \le k \le K-1$ and argmax is the set of policies that maximize the objective function (there might be multiple possible such optimal policies, hence the notation "$\in$ argmax").

6.1.3 Suboptimal Schemes

Finding optimal policies for (6.2) is notoriously intractable. Here, we are interested in the family of PDAO schemes to approximate the optimal solution, as introduced in the last section and formally defined below. First, let $\mathcal{X}^* = \mathcal{X} \cup \mathcal{X}^2 \cup \cdots$ denote the collection of all strings of states. Similarly, define $\mathcal{A}^* = \mathcal{A} \cup \mathcal{A}^2 \cup \cdots$. The basic idea is to introduce a function $f : \mathcal{X}^* \times \mathcal{A}^* \to \mathbb{R}_+$ such that at the horizon K, f is equivalent to g in the following sense: the string of policies $\pi_k^* : \mathcal{X} \to \mathcal{A}$, $k = 1, \ldots, K$, form an optimal solution to (6.2) if and only if it is also optimal for the objective function $\mathrm{E}[f(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K))|x_1]$. Then, at each intermediate state x_k, we simply optimize the function $f(x_1, \ldots, x_k; \pi_1(x_1), \ldots, \pi_{k-1}(x_{k-1}), \cdot)$ (with respect to its last action argument). We formalize these and other related concepts precisely below.

Note that we have explicitly distinguished between the objective function in terms of g, which is a function of K states and K actions, and the function f, which can take arguments with state and action strings that are of arbitrary length. The function f is what we introduce as a way to (approximately) solve problem (6.2) (i.e., g defines the given optimization problem while f defines our solution scheme to approximately solve (6.2).

We are now ready to define PDAO schemes formally. We assume throughout that $x_1 \in \mathcal{X}$ is given.

PDAO Scheme: The policy $(\pi_1^p, \ldots, \pi_K^p)$ is called a path-dependent action optimization (PDAO) solution if for $i = 1, \ldots, K$,

$$\pi_i^p(x_i^p) \in \operatorname*{argmax}_a f(x_1^p, \ldots, x_i^p; \pi_1^p(x_1^p), \ldots, \pi_{i-1}^p(x_{i-1}^p), a), \tag{6.3}$$

where $x_1^p = x_1$ is given and $x_{k+1}^p = h(x_k^p, \pi_k^p(x_k^p), \xi_k)$ for $1 \leq k \leq i-1$.

Note that we could have made f and h explicitly time dependent. However, time can always be incorporated into the state, and so our formulation is without loss of generality.

Next, we define another suboptimal scheme we call the greedy policy-selection scheme.

Greedy Policy-Selection Scheme (GPS): The policy $(\pi_1^g, \ldots, \pi_K^g)$ is called a greedy policy-selection (GPS) solution if for $i = 1, \ldots, K$,

$$\pi_i^g \in \underset{\pi_i}{\operatorname{argmax}} \, \mathrm{E}[f(x_1^g, \ldots, x_i^g; \pi_1^g(x_1^g), \ldots, \pi_{i-1}^g(x_{i-1}^g), \pi_i(x_i^g))|x_1], \tag{6.4}$$

where $x_1^g = x_1$ is given and $x_{k+1}^g = h(x_k^g, \pi_k^g(x_k^g), \xi_k)$ for $1 \leq k \leq i-1$.

Note that a PDAO scheme chooses a string of actions based on a particular sample path. On the other hand, a GPS scheme generates the policy mapping based on the expected reward. Nonetheless, a PDAO scheme still defines a particular policy.

6.1.4 Terminology and Definitions

In this section, we introduce some terminology and corresponding definitions that will be used throughout the paper.

Whenever we are given a policy $(\pi_1, \ldots, \pi_k)$ and we use the notation for states $x_1, x_2, x_3, \ldots$, we mean that these states satisfy the usual state transition law $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$.

Let Π be the set of all strings of policies $(\pi_1, \ldots, \pi_k)$ with $k = 0, 1, 2, \ldots$; the case $k = 0$ corresponds to the empty string. Given x_1, define the function $f_{\text{avg}} : \Pi \to \mathbb{R}_+$ by

$$f_{\text{avg}}(\pi_1, \ldots, \pi_k) = \mathrm{E}[f(x_1, \ldots, x_k; \pi_1(x_1), \ldots, \pi_k(x_k))|x_1].$$

It is clear that

$$f_{\text{avg}}(\pi_1, \ldots, \pi_K) = \mathrm{E}[f(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K))|x_1]$$

is the objective function in (6.2). So we have converted our original problem to one where the objective function f_{avg} is simply a function of policy strings. This allows us to bridge our original problem to one for which submodular optimization results apply. To complete this bridge, we will define the notion of submodularity formally as follows.

String-Submodularity: The function $f_{\text{avg}} : \Pi \to \mathbb{R}_+$ is string-submodular if the following properties hold:

i). **prefix-monotone** property: $\forall k = 1, \ldots, K$ and $(\pi_1, \pi_2, \ldots, \pi_k) \in \Pi$,

$$f_{\text{avg}}(\pi_1, \ldots, \pi_k) \geq f_{\text{avg}}(\pi_1, \ldots, \pi_{k-1}).$$

ii). **diminishing-return** property: $\forall k = 1, \ldots, K-1$ and $(\pi_1, \ldots, \pi_k, \hat{\pi}) \in \Pi$,

$$f_{\text{avg}}(\pi_1, \ldots, \pi_{k-1}, \hat{\pi}) - f_{\text{avg}}(\pi_1, \ldots, \pi_{k-1}) \geq f_{\text{avg}}(\pi_1, \ldots, \pi_k, \hat{\pi}) - f_{\text{avg}}(\pi_1, \ldots, \pi_k).$$

For convenience, henceforth we will simply use the term submodular to mean string-submodular.

6.2 Main Results

In this section, we first provide performance bounds for the GPS scheme in problem (6.2). Then we prove that any PDAO scheme is also a GPS scheme, so the results for GPS schemes can be used to bound PDAO schemes.

6.2.1 Performance Bounds for GPS

The following theorem provides performance bounds for the GPS scheme. This is the first step in our argument. Before we state the theorem, we review the notation from Section 6.1.3. We use $(\pi_1^*, \ldots, \pi_K^*)$ to denote an optimal policy for problem (6.2), and $(\pi_1^g, \ldots, \pi_K^g)$ to denote a GPS policy. The corresponding state sequences will also have the superscript $*$ and g, respectively. Recall that by virtue of the equivalence of f and g as defined earlier, $(\pi_1^*, \ldots, \pi_K^*)$ is also an optimal policy for f_{avg}.

Theorem 6.2.1. *Assume that $f_{avg} : \Pi \to \mathbb{R}_+$ is submodular and that for $k = 1, \ldots, K-1$,*

$$f_{avg}(\pi_1^g, \ldots, \pi_k^g, \pi_1^*, \ldots, \pi_K^*) \geq f_{avg}(\pi_1^*, \ldots, \pi_K^*).$$

Then, any GPS scheme $(\pi_1^g, \ldots, \pi_K^g)$ to problem (6.2) satisfies

$$\frac{f_{avg}(\pi_1^g, \ldots, \pi_K^g)}{f_{avg}(\pi_1^*, \ldots, \pi_K^*)} \geq 1 - \left(1 - \frac{1}{K}\right)^K > 1 - \frac{1}{e}. \tag{6.5}$$

The proof of the above theorem involves the following observations. First, we use the fact that $(\pi_1^*, \ldots, \pi_K^*)$ is also an optimal policy for f_{avg}. Second, we apply Theorem 1 in [4] in view of the assumptions on f_{avg} in the theorem.

The bound above can be improved by introducing the notion of curvature. We introduce below two notions of curvature that yield improved bounds,.

Definition 6.2.1. *The* ***total backward curvature*** *of $f_{avg} : \Pi \to \mathbb{R}_+$ is*

$$\sigma = 1 - \min_{\substack{(\pi_1,\ldots,\pi_k)\in\Pi \\ f_{avg}(\pi_1)\neq f_{avg}(\emptyset)}} \left\{ \frac{f_{avg}(\pi_1, \pi_2, \ldots, \pi_k) - f_{avg}(\pi_2, \ldots \pi_k)}{f_{avg}(\pi_1) - f_{avg}(\emptyset)} \right\},$$

Definition 6.2.2. *The* ***total backward curvature*** *of $f_{avg} : \Pi \to \mathbb{R}_+$ with respect to the optimal policy $(\pi_1^*, \ldots, \pi_K^*)$ by*

$$\sigma^* = 1 - \min_{\substack{(\pi_1,\ldots,\pi_k)\in\Pi \\ f_{avg}(\pi_1,\ldots,\pi_k)\neq f_{avg}(\emptyset)}} \left\{ \frac{f_{avg}(\pi_1, \ldots, \pi_k, \pi_1^*, \ldots, \pi_K^*) - f_{avg}(\pi_1^*, \ldots \pi_K^*)}{f_{avg}(\pi_1, \ldots, \pi_k) - f_{avg}(\emptyset)} \right\},$$

We will need another definition, an alternative notion of monotoneity.

Definition 6.2.3. *The function $f_{avg} : \Pi \to \mathbb{R}_+$ is* ***postfix-monotone*** *if $\forall k = 1, \ldots, K$ and $(\pi_1, \ldots, \pi_k) \in \Pi$,*

$$f_{avg}(\pi_1, \pi_2, \ldots, \pi_k) \geq f_{avg}(\pi_2, \ldots, \pi_k).$$

Notice the difference between the postfix-monotone and prefix-monotone properties.

Lemma 6.2.2. *If $f_{avg} : \Pi \to \mathbb{R}_+$ is postfix-monotone, then $\sigma^* \leq \sigma \leq 1$.*

The proof is straightforward and is omitted.

Theorem 6.2.3. *Assume that $f_{avg} : \Pi \to \mathbb{R}_+$ is submodular and postfix-monotone. Then, any GPS scheme $(\pi_1^g, \ldots, \pi_K^g)$ to problem (6.2) satisfies*

$$\frac{f_{avg}(\pi_1^g, \ldots, \pi_K^g)}{f_{avg}(\pi_1^*, \ldots, \pi_K^*)} \geq \frac{1}{\sigma^*}\left(1 - \left(1 - \frac{\sigma^*}{K}\right)^K\right) > \frac{1}{\sigma^*}\left(1 - e^{-\sigma^*}\right).$$

As in Theorem 6.2.1, the proof involves the fact that $(\pi_1^*, \ldots, \pi_K^*)$ is also an optimal policy for f_{avg} and applying Theorem 1 in [4].

From Lemma 1 and Theorem 2, the following holds.

Corollary 6.2.4. *Assume that $f_{avg} : \Pi \to \mathbb{R}_+$ is submodular and postfix-monotone. Then, any GPS scheme $(\pi_1^g, \ldots, \pi_K^g)$ to problem (6.2) satisfies*

$$\frac{f_{avg}(\pi_1^g, \ldots, \pi_K^g)}{f_{avg}(\pi_1^*, \ldots, \pi_K^*)} \geq \frac{1}{\sigma}\left(1 - \left(1 - \frac{\sigma}{K}\right)^K\right) > \frac{1}{\sigma}\left(1 - e^{-\sigma}\right).$$

6.2.2 Performance Bounds for PDAO

In this section, we will apply the results in Section 6.2.1 to derive performance bound for PDAO schemes. The key lies in the following theorem.

Theorem 6.2.5. *Any PDAO policy is also a GPS policy.*

Proof: Suppose that we are given a PDAO policy $(\pi_1^p, \ldots, \pi_K^p)$ (i.e., satisfying (6.3)). We will show that there exists a GPS policy $(\pi_1^g, \ldots, \pi_K^g)$ such that the two policies are equal. We will do this by showing that $\pi_j^p = \pi_j^g$ for $1 \leq j \leq k$ by induction on $k = 1, \ldots, K$.

For $k = 1$, by (6.3), we have that for any π_1,

$$f(x_1^p; \pi_1^p(x_1^p)) \geq f(x_1^p; \pi_1(x_1^p)), \tag{6.6}$$

which implies that

$$\mathrm{E}[f(x_1^p; \pi_1^p(x_1^p))|x_1] \geq \mathrm{E}[f(x_1^p; \pi_1(x_1^p))|x_1]. \tag{6.7}$$

Because $x_1^p = x_1$, this shows that π_1^p is also a GPS policy.

For the induction step, assume that there exists $(\pi_1^g, \ldots, \pi_k^g)$ satisfying (6.4) such that $\pi_j^p = \pi_j^g$ for $1 \leq j \leq k$. To complete the proof, it suffices to show that π_{k+1}^p satisfies (6.4).

By definition, we have that $x_{j+1}^p = h(x_j^p, \pi_j^p(x_j^p), \xi_j)$ and $x_{j+1}^g = h(x_j^g, \pi_j^g(x_j^g), \xi_j)$ for $1 \leq j \leq k$. Based on the assumption that $\pi_j^p = \pi_j^g$ for $1 \leq j \leq k$ and $x_1^p = x_1^g$, we have that $x_{j+1}^p = x_{j+1}^g$ for $1 \leq j \leq k$. Then we have that $x_{k+1}^p = x_{k+1}^g$.

For π_{k+1}^p, by (6.3), we have that for any π_{k+1},

$$\begin{aligned} &f(x_1^p, \ldots, x_{k+1}^p; \pi_1^p(x_1^p), \ldots, \pi_{k+1}^p(x_{k+1}^p)) \geq \\ &\quad f(x_1^p, \ldots, x_{k+1}^p; \pi_1^p(x_1^p), \ldots, \pi_{k+1}(x_{k+1}^p)), \end{aligned} \tag{6.8}$$

which implies that

$$\begin{aligned} &\mathrm{E}[f(x_1^p, \ldots, x_{k+1}^p; \pi_1^p(x_1^p), \ldots, \pi_{k+1}^p(x_{k+1}^p))|x_1] \geq \\ &\quad \mathrm{E}[f(x_1^p, \ldots, x_{k+1}^p; \pi_1^p(x_1^p), \ldots, \pi_{k+1}(x_{k+1}^p))|x_1]. \end{aligned} \tag{6.9}$$

Because $x_{k+1}^p = x_{k+1}^g$, this means that π_{k+1}^p satisfies (6.4). This completes our induction argument. □

Based on Theorems 6.2.1, 6.2.3, and 6.2.5, we have the following theorem, which provides performance bounds for PDAO schemes.

Theorem 6.2.6. *Assume that $f_{avg} : \Pi \to \mathbb{R}_+$ is submodular and that for $k = 1, \ldots, K-1$,*

$$f_{avg}(\pi_1^g, \ldots, \pi_k^g, \pi_1^*, \ldots, \pi_K^*) \geq f_{avg}(\pi_1^*, \ldots, \pi_K^*).$$

then any PDAO scheme $(\pi_1^p, \ldots, \pi_K^p)$ to problem (6.2) satisfies

$$\frac{f_{avg}(\pi_1^p, \ldots, \pi_K^p)}{f_{avg}(\pi_1^*, \ldots, \pi_K^*)} \geq 1 - \left(1 - \frac{1}{K}\right)^K > 1 - \frac{1}{e}.$$

If f_{avg} is postfix-monotone, then any PDAO scheme $(\pi_1^p, \ldots, \pi_K^p)$ to problem (6.2) satisfies

$$\frac{f_{avg}(\pi_1^p, \ldots, \pi_K^p)}{f_{avg}(\pi_1^*, \ldots, \pi_K^*)} \geq \frac{1}{\sigma}\left(1 - \left(1 - \frac{\sigma}{K}\right)^K\right) > \frac{1}{\sigma}(1 - e^{-\sigma}).$$

Theorem 6.2.6 provides conditions for the objective function in problem (6.2) such that PDAO schemes achieve some guaranteed performance bounds.

6.3 Application to Stochastic Optimal Control

6.3.1 Problem Statement

In this section, we consider the application of (6.2) to stochastic optimal control problems. In stochastic optimal control, the objective function has the following additive form:

$$\mathrm{E}[f(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K))|x_1] = \sum_{k=1}^{K} \mathrm{E}[r(x_k, \pi_k(x_k))|x_1],$$

where $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}+$ for $k = 1, \ldots, K$ is the immediate reward accrued at time k by applying π_k at state x_k, and $\sum_{k=1}^{K} \mathrm{E}[r(x_k, \pi_k(x_k))|x_1]$ denotes the conditional expected cumulative reward over a time horizon of length K given the initial state x_1. The stochastic optimal control problem can be written in the following form:

$$\begin{aligned} &\underset{\pi_1,\ldots,\pi_K}{\text{maximize}} \ \sum_{k=1}^{K} \mathrm{E}[r(x_k, \pi_k(x_k))|x_1] \\ &\text{s. t. } \ x_{k+1} = h(x_k, \pi_k(x_k), \xi_k), \ k = 1, \ldots, K-1. \end{aligned} \tag{6.10}$$

This problem also goes by the name Markov decision problem (MDP) (or Markov decision process), and arises in a wide variety of areas, including sensor resource management [75], congestion control [76], UAV guidance for multi-target tracking [14, 77, 78], and the game of Go [66].

6.3.2 Dynamic Programming

The solution to the stochastic optimal control problem above is characterized by Bellman's principle of dynamic programming. To explain, for each $k = 1, \ldots, K$, define functions $V_k : \mathcal{X} \times \Pi_k \to \mathbb{R}$ by

$$V_k(x_k, \pi_k, \ldots, \pi_K) = \sum_{i=k}^{K} \mathrm{E}[r(x_i, \pi_i(x_i))|x_k]$$

where Π_k denotes the set of all strings $(\pi_k, \ldots, \pi_K)$ for $k = 1, \ldots, K$ and $x_{i+1} = h(x_i, \pi_i(x_i), \xi_i)$, $i = k, \ldots, K-1$. The objective function of problem (6.10) can be written as

$$V_1(x_1, \pi_1, \ldots, \pi_K)$$

where $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k),\ k = 1, \ldots, K-1$.

As before, let $\pi_1^*, \ldots, \pi_K^*$ be an optimal solution to problem (6.10), and given x_1, define $x_1^* = x_1$ and $x_{k+1}^* = h(x_k^*, \pi_k^*(x_k^*), \xi_k)$, $k = 1, \ldots, K-1$. Then, Bellman's principle states that for $k = 1, \ldots, K$,

$$\begin{aligned} &V_k(x_k^*, \pi_k^*, \ldots, \pi_K^*) \\ &= \max_{a \in \mathcal{A}} \{r(x_k^*, a) + \mathrm{E}[V_{k+1}(x_{k+1}^a, \pi_{k+1}^*, \ldots, \pi_K^*)|x_k^*, a]\}, \\ &\pi_k^*(x_k^*) \in \operatorname*{argmax}_{a \in \mathcal{A}} \{r(x_k^*, a) + \mathrm{E}[V_{k+1}(x_{k+1}^a, \pi_{k+1}^*, \ldots, \pi_K^*)|x_k^*, a]\} \end{aligned} \tag{6.11}$$

where $x_{k+1}^a = h(x_k^*, a, \xi_k)$ and $x_{i+1}^a = h(x_i^a, \pi_i^*(x_i^a), \xi_i)$ for $i = k+1, \ldots, K-1$, with the convention that $V_{K+1}(\cdot) \equiv 0$. Moreover, any policy satisfying (6.11) above is optimal. The term $\mathrm{E}[V_{k+1}(x_{k+1}^a, \pi_{k+1}^*, \ldots, \pi_K^*)|x_k^*, a]$ is called the expected value-to-go (EVTG).

Bellman's principle provides a method to compute an optimal solution: We use (6.11) to iterate backwards over the time indices $k = K, K-1, \ldots, 1$, keeping the states as variables, working all the way back to $k = 1$. This is the familiar dynamic programming algorithm. However, the procedure suffers from the curse of dimensionality [48] and is therefore impractical for many problems of interest. Therefore, designing computationally tractable approximation methods remains a topic of active research.

6.3.3 Approximate Dynamic Programming

In this section, we will discuss a class of schemes to approximate the optimal solution based on Bellman's principle and show that these are all PDAO schemes. The class of approximate dynamic programming (ADP) schemes rests on approximating the EVTG $\mathrm{E}[V_{k+1}(x^a_{k+1}, \pi^*_{k+1}, \ldots, \pi^*_K)|x^*_k, a]$ by some other term $W_{k+1}(\hat{x}_k, a)$. In this method, we start at time $k = 1$, at state $\hat{x}_1 = x_1$, and for each $k = 1, \ldots, K$, we compute the subsequent control actions and states using

$$\begin{aligned} &\hat{\pi}_k(\hat{x}_k) \in \operatorname*{argmax}_{a \in \mathcal{A}} \{r(\hat{x}_k, a) + W_{k+1}(\hat{x}_k, a)\} \\ &\text{and } \hat{x}_{k+1} = h(\hat{x}_k, \hat{\pi}_k(\hat{x}_k), \xi_k). \end{aligned} \tag{6.12}$$

The EVTG approximation term $W_{k+1}(\hat{x}_k, a)$ can be based on a number of methods, ranging from heuristics to reinforcement learning [79] to rollout [47].

When

$$W_{k+1}(\hat{x}_k, a) = \mathrm{E}[V_{k+1}(x^a_{k+1}, \pi^*_{k+1}, \ldots, \pi^*_K)|\hat{x}_k, a],$$

the ADP scheme is optimal. When $W_{k+1}(\hat{x}_k, a) = 0$, the ADP scheme is the myopic heuristic. When

$$W_{k+1}(\hat{x}_k, a) = \mathrm{E}\left[\sum_{i=k+1}^{K} r_i(\hat{x}_i, \pi_b(\hat{x}_i)) \middle| \hat{x}_k, a\right],$$

where $\hat{x}_1 = x_1$, $\hat{x}_{i+1} = h(\hat{x}_i, \hat{\pi}_i(\hat{x}_i), \xi_i)$ for $i = 1 \ldots, k$, and $\hat{x}_{i+1} = h(\hat{x}_i, \pi_b(\hat{x}_i), \xi_i)$ for $i = k+1 \ldots, K-1$, this ADP scheme is called rollout, and the policy π_b is called the base policy.

What is the performance of an ADP scheme above relative to the optimal solution? The answer, of course, depends on the specific EVTG approximation. If the EVTG approximation is equal to the true EVTG, then the procedure above generates an optimal solution. In general, the procedure produces something suboptimal. But how suboptimal? This question has alluded general treatment but has remained an issue of great interest to designers and users of ADP methods.

We address this issue using our framework of bounding PDAO schemes. More specifically, our idea is to formulate a stochastic optimization problem such that the ADP procedure above reduces to a PDAO scheme. Then, contingent on showing that submodularity and curvature conditions hold, our framework for bounding ADP schemes provides a systematic means to bound the performance of the ADP method.

To see how our approach works, define the function $f : \mathcal{X}^* \times \mathcal{A}^* \to \mathbb{R}_+$ by

$$f(x_1, \ldots, x_k; \pi_1(x_1), \ldots, \pi_k(x_k)) = \sum_{i=1}^{k} r_i(x_i, \pi_i(x_i)) + W_{k+1}(x_k, \pi_k(x_k)),$$

where $k = 1, \ldots, K$, $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$ as before, and $W_{K+1}(\cdot) \equiv 0$ by convention. Using this function f, we now have an associated PDAO scheme.

It is clear that at the terminal $k = K$, by the definition of f_{avg} in Section 6.1.4, we have that

$$f_{avg}(\pi_1, \ldots, \pi_K) = \mathrm{E}[f(x_1, \ldots, x_K; \pi_1(x_1), \ldots, \pi_K(x_K))|x_1] = \sum_{i=1}^{K} \mathrm{E}[r_i(x_i, \pi_i(x_i))|x_1],$$

which is equal to the objective function for the given problem (6.10), also the function to be maximized at the final stage for GPS scheme. By Theorem 6.2.5, we have that any PDAO policy is a GPS policy, which implies that we have established the equivalence of our f with the given problem. Next, notice that the PDAO scheme by definition has the following form, given $\pi_1(x_1), \ldots, \pi_{k-1}(x_{k-1})$:

$$\begin{aligned} \pi_k(x_k) &\in \operatorname*{argmax}_{a \in \mathcal{A}} f(x_1, \ldots, x_k; \pi_1(x_1), \ldots, \pi_{k-1}(x_{k-1}), a) \\ &= \operatorname*{argmax}_{a \in \mathcal{A}} \{\sum_{i=1}^{k-1} r(x_i, \pi_i(x_i)) + r(x_k, a) + W_{k+1}(x_k, a)\} \\ &= \operatorname*{argmax}_{a \in \mathcal{A}} \{r(x_k, a) + W_{k+1}(x_k, a)\}. \end{aligned}$$

But this is simply the ADP scheme in (6.12). Hence, we have the following result.

Proposition 6.3.1. *The ADP scheme in (6.12) is a PDAO scheme for the optimization problem defined above.*

6.3.4 Bounding ADP schemes

Our results for bounding PDAO schemes provide the basis for designing good ADP methods, namely by designing the approximate EVTG term to make the objective function satisfy the assumptions of Theorem 6.2.1 (and possibly Theorem 6.2.3 too). We now discuss how to satisfy these requirements.

From the last section, we have that for $k = 1, \ldots, K$,

$$f(x_1, \ldots, x_k; \pi_1(x_1), \ldots, \pi_k(x_k)) = \sum_{i=1}^{k} r(x_i, \pi_i(x_i)) + W_{k+1}(x_k, \pi_k(x_k)), \tag{6.13}$$

where $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$ as before and $W_{K+1}(\cdot) \equiv 0$. Then we have that

$$f_{\text{avg}}(\pi_1, \ldots, \pi_k) = \sum_{i=1}^{k} \mathrm{E}[r(x_i, \pi_i(x_i))|x_1] + \mathrm{E}[W_{k+1}(x_k, \pi_k(x_k))|x_1].$$

By the definition of the prefix-monotone property, we have the following sufficient conditions for f_{avg} to be prefix-monotone: $\forall x_k, \pi_k, \pi$,

$$\mathrm{E}[(r(x_{k+1}, \pi(x_{k+1})) + W_{k+2}(x_{k+1}, \pi(x_{k+1})))|x_k] \geq \mathrm{E}[W_{k+1}(x_k, \pi_k(x_k))|x_k], \tag{6.14}$$

where, as usual, $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$.

By the definition of diminishing-return property, we have the following sufficient conditions for f_{avg} to satisfy the diminishing-return property: $\forall x_k, \pi_k, \pi_{k+1}, \pi$,

$$\mathrm{E}[(r(x_{k+1}, \pi(x_{k+1})) + W_{k+2}(x_{k+1}, \pi(x_{k+1})) - W_{k+1}(x_k, \pi_k(x_k)))|x_k] \geq$$
$$\mathrm{E}[(r(x_{k+2}, \pi(x_{k+2})) + W_{k+3}(x_{k+2}, \pi(x_{k+2})) - W_{k+2}(x_{k+1}, \pi_{k+1}(x_{k+1})))|x_k]$$

where $x_{k+1} = h(x_k, \pi_k(x_k), \xi_k)$ and $x_{k+2} = h(x_{k+1}, \pi_{k+1}(x_{k+1}), \xi_{k+1})$.

By (6.13), we have that the objective function for ADP schemes is defined for strings of length up to K, so in order to satisfy the conditions $f_{avg}(\pi_1^g, \ldots, \pi_k^g, \pi_1^*, \ldots, \pi_K^*) \geq f_{avg}(\pi_1^*, \ldots, \pi_K^*)$ for $k = 1, \ldots, K-1$ in Theorem 6.2.1, we have to define f for strings of length greater than K. The problem of extending a string-submodular function defined for strings of length up to K to one defined for strings of length up to $2K-1$ is one of our future research directions. In Chapter 5, we provided conditions involving strings of length up to K for the greedy strategy to achieve the bound $1-(1-1/K)^K$, so we can consider similar conditions in order to avoid the above problem.

Chapter 7

Conclusion Summary

In Chapter 2, we developed bounds on the performance of the batched greedy strategy relative to the optimal strategy in terms of a parameter called the total batched curvature. We showed that when the objective function is a polymatroid set function, the batched greedy strategy satisfies a harmonic bound for a general matroid constraint and an exponential bound for a uniform matroid constraint, both in terms of the total batched curvature. We also studied the behavior of the bounds as functions of the batch size. Specifically, we proved that the harmonic bound for a general matroid is nondecreasing in the batch size and the exponential bound for a uniform matroid is nondecreasing in the batch size under the condition that the batch size divides the rank of the uniform matroid. Finally, we illustrated our results by considering a task scheduling problem and an adaptive sensing problem.

A related problem setting in one where the argument of the objective function is not a set but an ordered tuple, called a string. The performance of the 1-batch greedy strategy for string optimization problem has been investigated in [3, 4]; however, the performance of the general k-batch greedy strategy for string optimization has not been investigated so far. As was the case in Chapter 2, lifting does not work in the string setting. Moreover, batching in string submodular functions does not preserve submodularity in general. This makes analyzing the k-batch greedy strategy for string problems more challenging than for set problems, and remains open to date.

In Chapter 3, suppose that a function f defined on a matroid $(X, \mathcal{I})$ is extendable to the entire power set. We have shown that the majorizing extension algorithm does not always successfully produce this extension. Next, we explored defining a notion of curvature $b(f)$ depending only on sets in the matroid $(X, \mathcal{I})$, and we asked if it is always possible to extend f to g in such a way that $c(g) = b(f)$. Here, again, we have shown that the answer is in general negative; we gave necessary and sufficient conditions for $c(g) = b(f)$. This leaves us with the following ultimate question: What extension g of f has the best (smallest) value of $c(g)$? Unfortunately, answering this question boils down to solving an optimization problem that is in general as difficult as (3.1), solvable using only something like dynamic programming. This, of course, does not point to a practical algorithm for finding an extension with the best curvature.

In Chapter 4, we considered variations of the non-cooperative utility system considered by Vetta, in which users are grouped together. We considered two types of grouping among users in utility systems. The first type of grouping is from [2], where each user belongs to a group of users having social ties with it. For this type of utility system, each user takes its strategy by maximizing its social group utility function, giving rise to the notion of social-aware Nash equilibrium. We proved that this social utility system yields to the bounding results of Vetta for non-cooperative system, thus establishing provable performance guarantees for the social-aware Nash equilibria. For the second type of grouping we considered, the set of users is partitioned into l disjoint groups, where the users within a group takes their group strategy by maximizing their group utility, giving rise to the notion of the group Nash equilibrium. In this case, each group can be viewed as a new user with vector-valued actions, and a 1/2 bound for the performance of group Nash equilibria follows from the result of [1]. By defining the group curvature c_{k_i} associated with group i with k_i users, we showed that if the social utility function is nondecreasing and submodular, then any group Nash equilibrium achieves at least $1/(1 + \max_{1\leq i\leq l} c_{k_i})$ of the optimal social utility. Especially, if each user has the same action space, then we showed that any group Nash equilibrium achieves at least $1/(1 + c_{k^*})$ of the optimal social utility, where k^* is the least number of users among the l groups.

In Chapter 5, we considered an optimization problem where the decision variable is a string of length at most K. For this problem, we reviewed some previous results on bounding the greedy strategy. In particular, the results of [4] provide sufficient conditions for the greedy strategy to be bounded by a factor of $(1-(1-1/K)^K)$. We then presented weakened sufficient conditions for this same bound to hold, by introducing the notions of K-submodularity and K-GO-concavity. Next, we introduced a notion of curvature $\eta \in (0, 1]$, which furnishes an even tighter bound with the factor $\frac{1}{\eta}(1 - e^{-\eta})$. Finally, we illustrated our results by considering two example applications. We showed that our new results provide weaker conditions on parameter values in these applications than in [4]. Finally, we presented an example of a utility system for database assisted spectrum access to illustrate our results.

In Chapter 6, we developed a framework to bound the performance of path-dependent action optimization (PDAO) schemes. We showed that every PDAO scheme is a greedy scheme for some optimization problem, and if that optimization problem is equivalent to our problem of interest and is provably submodular, then we can say that our PDAO scheme is no worse than something like $(1 - e^{-1})$ of optimal. We demonstrated how our framework can be applied in stochastic optimal control problems to systematically bound the performance of general approximate dynamic programming (ADP) schemes. The question how to design an ADP scheme such that it satisfies our framework in real applications remains one of our future research directions.

Bibliography

[1] A. Vetta. Nash equilibrium in competitive societies, with applications to facility location, traffic routing and auctions. *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 416–425, 2002.

[2] X. Chen, X. Gong, L. Yang, and J. Zhang. A social group utility maximization framework with applications in database assisted spectrum access. *The 33rd Annual IEEE International Conference on Computer Communications (INFOCOM'14)*, pages 1959–1967, 2014.

[3] M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'08)*, pages 1577–1584, 2008.

[4] Z. L. Zhang, E. K. P. Chong, A. Pezeshki, and W. Moran. String submodular functions with curvature constraints. *IEEE Trans. Autom. Control*, 61:601–616, 2016.

[5] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions–II. *Math. Program. Stud.*, 8:73–87, 1978.

[6] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–I. *Math. Program.*, 14:265–294, 1978.

[7] L. Lovász. Submodular functions and convexity. *Mathematical Programming The State of the Art*, pages 235–257, 1983.

[8] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 137–146, 2003.

[9] A. Krause, A. Singh, and C. Guestrin. *Near-Optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies*. 2014.

[10] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. *International Conference on Computer Vision (ICCV)*, pages 105–112, 2001.

[11] H. Lin and J. Bilmes. A class of submodular functions for document summarization. volume 1, pages 510–520. Association for Computational Linguistics, 2011.

[12] Y. Li, K. Wei, K. Kirchhoff, Y. Song, and J. Bilmes. Submodular feature selection for high-dimensional acoustic score spaces. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7184–7188, 2013.

[13] K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. *Proceedings of the 32Nd International Conference on International Conference on Machine Learning (ICML'15)*, 37:1954–1963, 2015.

[14] S. Ragi and E. K. P. Chong. UAV path planning in a dynamic environment via partially observable Markov decision process. *IEEE Transactions on Aerospace and Electronic Systems*, 49:2397–2412, 2013.

[15] A. Krause and D. Golovin. Submodular function maximization. *In Tractability: Practical Approaches to Hard Problems. Cambridge University Press*, 2014.

[16] U. Feige and J. Vondrák. Approximation algorithms for allocation problems: Improving the factor of $1 - 1/e$. *In: Proc. of 17th Annual ACM-SIAM Symp. Discrete Algorithm*, pages 667–676, 2006.

[17] L. Fleischer, M.X. Goemans, V.S. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. *In: Proc. of 17th Annual ACM-SIAM Symp. Discrete Algorithm*, pages 611–620, 2006.

[18] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Inf. Process. Lett.*, 100:162–166, 2006.

[19] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.

[20] G. Calinescu, C. Chekuri, and Vondrák J. Pál, M. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40:1740–1746, 2011.

[21] N. Korula, V. Mirrokni, and Zadimoghaddam M. Online submodular welfare maximization: greedy beats 1/2 in random order. *The 47th Annual Symp. on Theory of Computing*, pages 889–898, 2015.

[22] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 67–74, 2008.

[23] D. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Nav. Res. Log.*, 45:615–627, 1998.

[24] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45:634–652, 1998.

[25] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70:39–45, 1999.

[26] G. Cornuéjols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, 1977.

[27] N. Megiddo, E. Zemel, and S.L. Hakimi. The maximum coverage location problem. *SIAM J. Alg. Disc. Math.*, 4:253–261, 1983.

[28] R. Church and C. R. Velle. The maximal covering location problem. *Papers in Regional Science*, 32:101–118, 1974.

[29] H. Pirkul and D.A. Schilling. The maximal covering location problem with capacities on total workload. *Manag. Sci.*, pages 233–248.

[30] E. Liu, E. K. P. Chong, and L. L. Scharf. Greedy adaptive linear compression in signal-plus-noise models. *IEEE Trans. Inf. Theory*, 60:2269–2280, 2014.

[31] Y. Chen, C. Chuah, and Q. Zhao. Sensor placement for maximizing lifetime per unit cost in wireless sensor networks. *MILCOM 2005 - 2005 IEEE Military Communications Conference*, pages 1097–1102, 2005.

[32] S. Ragi, H.D. Mittelmann, and E. K. P. Chong. Directional sensor control: Heuristic approaches. *IEEE Sens. J.*, 15:374–381, 2015.

[33] D. Hausmann, B. Korte, and T. A. Jenkyns. Worst case analysis of greedy type algorithms for independence systems. *Math. Program. Stud.*, 12:120–131, 1980.

[34] M. Conforti and G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Appl. Math.*, 7:251–274, 1984.

[35] J. Vondrák. Submodularity and curvature: The optimal algorithm. *RIMS Kokyuroku Bessatsu*, B23:253–266, 2010.

[36] M. Sviridenko, J. Vondrák, and J. Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Procedings of the 26th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 1134–1148, 2015.

[37] S. Ahmed and A. Atamtürk. Maximizing a class of submodular utility functions. *Mathematical Programming*, 128(1-2):149–169, 2011.

[38] G. Arslan, J. R. Marden, and J. S. Shamma. Autonomous vehicle-target assignment: a game-theoretical formulation. *J. Dyn. Syst. Meas. Control*, 129:584–596, 2007.

[39] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-noy, T. Brown, and T. L. Porta. A survey of sensor selection schemes in wireless sensor networks. *Proceedings of SPIE 2007*, 6562, 2007.

[40] B. Allen, G. Lippner, Y. T. Chen, B. Fotouhi, N. Momeni, S. T. Yau, and M. A. Nowak. Evolutionary dynamics on any population structure. *Nature*, 544:227–230, 2017.

[41] M. L. Littman. Algorithms for sequential decision making. *Ph.D. thesis, Department of Computer Science, Brown University*, 1996.

[42] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[43] S. Nozhati, Y. Sarkale, B. Ellingwood, E. K. P. Chong, and H. Mahmoud. Near-optimal planning using approximate dynamic programming to enhance post-hazard community resilience management. *arXiv:1803.01451*, 2018.

[44] E. Liu, E. K. P. Chong, and L. L. Scharf. Greedy adaptive linear compression in signal-plus-noise models. *IEEE Trans. Inf. Theory*, 60:2269–2280, 2014.

[45] R. A. Jarvis. Optimization strategies in adaptive control: a selective survey. *IEEE Trans. Syst. Man. Cybern.*, SMC-5:83–94, 1975.

[46] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt. Dynamic optimization using adaptive control vector parameterization. *Computers & Chemical Engineering*, 29:1731–1751, 2005.

[47] D. P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, third edition, 2005.

[48] W. B. Powell. *Approximate dynamic programming: solving the curses of dimensionality*. New York: Wiley, 2007.

[49] E. K. P. Chong, C. Kreucher, and A. O. Hero. Partially observable Markov decision process approximations for adaptive sensing. *Discrete Event Dynamic Systems*, 19:377–422, 2009.

[50] Y. Liu, Z. Zhang, E.K.P. Chong, and A. Pezeshki. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Proc. of 2016 American Control Conf.*, pages 7177–7182, 2016.

[51] D.P. Chandu. Big step greedy heuristic for maximum coverage problem. *Int. J. Comput. Appl.*, 125(7):19–24, 2015.

[52] J. Lee, M. Sviridenko, and J. Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35:795–806, 2010.

[53] Y. J. Liu, Z. L. Zhang, E. K. P. Chong, and A. Pezeshki. Performance Bounds with Curvature for Batched Greedy Optimization. *J. Optim. Theory Appl.*, 177:535–562, 2018.

[54] J. Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial Optimization*, 2570:11–26, 2003.

[55] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. An inequality for polymatroid functions and its applications. *Discrete Appl. Math.*, 131(2):255–281, 2003.

[56] W.T. Tutte. Lecture on matroids. *J. Res. Natl. Bur. Stand.-Sec. B: Math. Math. Phys.*, 69B:1–47, 1965.

[57] Y. J. Liu, Chong. E. K. P., and A. Pezeshki. Bounding the greedy strategy in finite-horizon string optimization. *The 54th IEEE Conference on Decision and Control (CDC)*, pages 3900–3905, 2015.

[58] Y. J. Liu, E. K. P. Chong, and A. Pezeshki. Extending polymatroid set functions with curvature and bounding the greedy strategy. *Proceedings of the 2018 IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

[59] Y. J. Liu, E. K. P. Chong, and A. Pezeshki. Improved bounds for the greedy strategy in optimization problems with curvature, 2018.

[60] A.P. Singh, A. Guillory, and J. Bilmes. On bisubmodular maximization. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 1055–1063, 2012.

[61] Y. J. Liu, E. K. P. Chong, and A. Pezeshki. Performance Bounds for Nash Equilibria in Submodular Utility Systems with User Groups. *Journal of Control and Decision*, 5:1–18, 2017.

[62] S. T. Jawaid and S. L. Smith. Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*, 61:282–288, 2015.

[63] S. Alaei and A. Malekian. Maximizing sequence-submodular functions and its application to online advertising. *arXiv preprint arXiv:1009.4153*, 2010.

[64] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[65] Z. Wang, W. Moran, X. Wang, and Q. Pan. Approximation for maximizing monotone non-decreasing set functions with a greedy method. *Journal of Combinatorial Optimization*, 31:29–43, 2014.

[66] D. Silver, Huang A., C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[67] Y. J. Liu, Chong. E. K. P., A. Pezeshki, and B. Moran. Bounds for approximate dynamic programming based on string optimization and curvature. *The 53rd IEEE Conference on Decision and Control (CDC)*, pages 6653–6658, 2014.

[68] G. L. Nemhauser and L. A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3:177–188, 1978.

[69] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43, 2004.

[70] U. Feige and J. Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6:247–290, 2010.

[71] M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: Leveraging submodularity. *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 2572–2577, 2010.

[72] A. A. Ageev and M. I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8:307–328, 2004.

[73] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. *Proc. 20th ACM-SIAM Symposium on Discrete Algorithms*, pages 545–554, 2009.

[74] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

[75] A. O. Hero, D. Castanon, D. Cochran, and K. Kastella. *Foundations and Applications of Sensor Management*. New York, NY: Springer, 2008.

[76] G. Wu, E. K. P. Chong, and R. L. Givan. Burst-level congestion control using hindsight optimization. *IEEE Trans. Autom. Control.*, 47:979–991, 2002.

[77] S. A. Miller, Z. A. Harris, and E. K. P. Chong. A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking. *EURASIP Journal on Applied Signal Processing*, 2009:1–17, 2009.

[78] S. Ragi and E. K. P. Chong. Decentralized guidance control of UAVs with explicit optimization of communication. *Journal of Intelligent and Robotic Systems*, 73:811–822, 2014.

[79] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.