Incentives, Computation, and Networks: Limitations and Possibilities of Algorithmic Mechanism Design

by

Yaron Singer

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Christos Papadimitriou, Chair
Professor Shachar Kariv
Professor Scott Shenker

Spring 2012

Incentives, Computation, and Networks: Limitations and Possibilities of Algorithmic Mechanism Design

Abstract

Incentives, Computation, and Networks: Limitations and Possibilities of Algorithmic Mechanism Design

by

Yaron Singer

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Christos Papadimitriou, Chair

In the past decade, a theory of manipulation-robust algorithms has been emerging to address the challenges that frequently occur in strategic environments such as the internet. The theory, known as algorithmic mechanism design, builds on the foundations of classical mechanism design from microeconomics and is based on the idea of incentive compatible protocols. Such protocols achieve system-wide objectives through careful design that ensures it is in every agent's best interest to comply with the protocol. As it turns out, however, implementing incentive compatible protocols as advocated in classical mechanism design theory often necessitates solving intractable problems. To address this, algorithmic mechanism design focuses on designing computationally-feasible incentive compatible approximation algorithms.

In the first part of this thesis we show the limitations of algorithmic mechanism design. We introduce a novel class of problems which are approximable in the absence of strategic constraints, and have an optimal incentive compatible solution when no computational constraints are enforced; we show that, under standard computational assumptions, for this class of problems there is no algorithm with a reasonable approximation ratio that is both computationally feasible and incentive compatible. This settles the central open question in algorithmic mechanism design which, since its inception, has been focused on trying to show the hardness of polynomial time incentive compatibility.

In the second part of this thesis we show the possibilities of algorithmic mechanism design. We introduce a novel class of problems where the bottleneck for implementation is the constraint on payments. We show that for a broad class of these problems, there are incentive compatible mechanisms with desirable approximation guarantees that do not require overpayments. By resulting to approximations, this result circumvents well known impossibility results from classical mechanism design theory that deem incentive compatibility to be infeasible under a budget.

To my parents.

Contents

Acknowledgments

I would like to thank Christos for his devoted parenting during my academic adolescence. Despite my best efforts, I was not able to discover the boundaries of his patience, generosity, and encouragement, and I am grateful to him for exercising these qualities with me. If I had to summarize the most important thing I learned from Christos in a single sentence I would say that I learned that in the limit, there is no point in trying to optimize the constants. If I had to summarize it in two sentences I would probably add something clever about truthfulness. I couldn't have wished for a better mentor.

I thank Scott Shenker for his guidance throughout my time in Berkeley and most importantly for believing in me. I thank Shachar Kariv for many passionate discussions that made for invaluable contributions to my Economics education. I wish to also thank the faculty and students of the theory group and the RAD (AMP) Lab, as well as David Ahn, Shachar, Chris Shanon, and Adam Szeidl from the Economics department, all of whom made speeding on interdisciplinary bridges possible.

I would like to thank my fabulous collaborators. In particular, I am greatly thankful for the friendship and support of Michael Schapira with whom I had the privilege to work on topics discussed in the first part of this thesis. I thank Moshe Babaioff and Liad Blumrosen for mentoring me during my summer at the wonderful Microsoft Research lab in Silicon Valley. I wish to thank Bobby Kleinberg and Èva Tardos for hosting me at Cornell and for our fruitful discussions directly related to the work in this thesis. I also warmly thank Jake Abernethy, Ashwinkumar Badanidiyuru, Dave Buchfuhrer, Joan de Marti, Ilias Diakonikolas, Shachar Dobzinski, Shaddin Dughmi, Joan Feigenbaum, Eric Friedman, Manas Mittal, Elchanan Mossel, George Pierrakos, Yuval Shavitt, Christos Stergiou, and Vasilis Syrganis.

I thank Microsoft Research and Facebook for their generous support during my studies. These votes of confidence came at times when I needed them most, and gave me the freedom to explore new territories, some of which are beyond the scope of this thesis.

This thesis is a summary of four magical years in Berkeley. If you read the proofs carefully, you'll smell the coffee beans of Yali's, Nefeli's and PiQ. I wish to thank Ayal, Nasos, and Rufo, and the talented baristas of these special places for their caffeine and hospitality.

I thank my sister Maya for advocating for me from the day I was born. I thank Yair for being a mensch and for, together with Maya, making the Bay Area home to us. I thank Arielli and Karini for melting my heart with their laughter. Every single time.

I remember asking my mother when I was six whether I'll have to get a PhD when I grow up. When she laughed and said I don't, I engraved that moment in my memory, to have an argument just in case she ever changed her mind. I thank my parents for letting me explore my own paths, even when they feared for my life or that I'll become a businessman. I thank them for their love and for being so proud.

And to Meromit, what I write here is insignificant considering what you have given me. Thank you for making this thesis; for making my deadlines your own, sharing my anxieties and small victories; for celebrating with me every time I said I'm basically done writing this thesis, and for the next time you'll celebrate with me. For always being that girl I once met.

Chapter 1

Introduction

1.1 The Internet

The rapid emergence of the internet and the world wide web have changed the way we think about markets and computation. From network protocols between Internet Service Providers (ISPs) to large-scale search engine keyword auctions, the internet is rich with examples where multiple – often strategic – entities with competing objectives interact in a computational environment. As a result, in the past decade algorithms have been fortified with concepts like equilibrium and incentives, and computational thought has been injected into market design.

Almost by its definition, the internet is an arena of strategic interactions. It is a computational artifact – first of its kind – in which resources are owned by strategic agents, distributed across an evolving networked system. The actual transmission of data in the internet is a coordinated effort by multiple ISPs bound via various business relationships that transfer data packets between them. Each such ISP has its own objectives to optimize, and may strategically misreport its available resources or relationships with other ISPs to satisfy its own goals. To achieve system-wide objectives, it is therefore not enough for algorithms to be computationally feasible; they must also robust to manipulation by entities that provide their input.

From a market design perspective, as the world wide web becomes the main medium for consuming and sharing information, markets are moving online where achieving economic efficiency is conditioned on computational tractability. A useful example is that of advertisement auctions which are currently the leading monetization model for content on the web. Advertisement space in web pages is allocated via auctions that take bids from agents who wish to place advertisements that are relevant to the page content. Advertisement platforms allow bidding agents to express complex preferences and typically conduct millions of auctions in a single minute. Naturally, efficient computation is at the heart of these auctions and requires supplementing traditional economics with optimization machinery under limited computational resources.

In the last several years the internet is going through yet another change. It is becoming social. The recent developments in web technologies and the emergence of online social networks enable people to easily create and share content across the web. This coming generation of internet markets aims at creating effective recommendation systems and marketing platforms by leveraging petabytes of online social interaction data. Incentives, computation, and network effects all play a major role in the social web and present some of its greatest problems and opportunities.

The challenges of the internet therefore call for an *algorithmic game theory* [47]: a fusion of ideas from mathematical economics, game theory and computer science to develop algorithms capable of optimizing complex market objectives in strategic environments.

1.2 Algorithmic Mechanism Design

In microeconomics, a theory for protocol design in strategic environments has been developed for the past fifty years in the field known as *mechanism design*. The theory is based on the idea of *incentive compatible* (or *truthful*) protocols that guarantee it is in every participating agent's best interest to act according to prescribed rules, which enables achieving global system objectives. In the past decade, computational considerations and algorithmic techniques have been incorporated into mechanism design, introducing *algorithmic mechanism design* [45]. This field builds on the foundations of traditional mechanism design and incorporates models and concepts from the theory of computation to develop a theory of algorithms for strategic environments. At a high level, algorithmic mechanism design can be categorized by two main efforts:

- **Limitations.** Algorithms that are robust to manipulation are naturally more limited than algorithms that are not. Since its inception, algorithmic mechanism design has been focused on identifying an example that shows that there is a substantial gap in the performance of incentive compatible algorithms and unrestricted algorithms. Once established, a canonical impossibility result enables exploring alternative (weaker but feasible) solution concepts that can be implemented instead.

- **Possibilities.** Designing markets where optimization of market objectives must be aligned with agents' incentives requires developing new algorithmic machinery. Algorithmic mechanism design considers computational limitations and develops general techniques for designing computationally feasible mechanisms. Moreover, even in cases where computation is not a limitation, the field often introduces machinery from optimization and combinatorics that enable addressing the complexities of online markets.

Naturally, there are strong intersections and overlaps between these directions: Algorithms draw the boundaries of computational and game theoretic limitations and vice versa. The above categorization of research threads, while somewhat artificial, is useful in narrating the main results presented in this thesis.

In this thesis we introduce two novel models that shed light on the limitations and possibilities of algorithmic mechanism design. Both models feed off the richness and complexity in reconciling incentives and computation in real-world networks. The first model is the *combinatorial public projects* which captures preference aggregation in networks and settles the decade-long open question on the computational hardness of incentive compatibility. The *budget feasibility* model described in the second part of this thesis was developed for optimizing influence in social networks and provides powerful machinery for designing procurement markets. We briefly summarize of the main points and motivation behind these models.

1.3 Part I: Limitations of Algorithmic Mechanism Design

Mechanism design theory has been well established in microeconomics for the past fifty years and is based on the idea of incentive compatibility: protocols for which it is provably in every participating agent's interest to report their information truthfully. The idea is simple and brilliant: the fact that truthful reporting is the optimal strategy guarantees that the input that an incentive compatible algorithm receives from rational agents is valid, and the quality of its solution can be compared against the real optimum. Incentive compatibility often requires payments, and the term *mechanism* is used to describe a pair of carefully designed algorithm and payment rule that dictate an allocation and payment for the participating agents.

A fundamental result in this field is the celebrated VCG mechanism [60, 16, 30] which gives a general recipe for designing incentive compatible protocols that yield optimal solutions and for years has been considered a certificate for the wide-range applicability of incentive compatibility. In the past decade however, with the emergence of internet-scale markets, a strong doubt has been cast over the generality of this result: To be applicable, VCG requires computing the *exact* optimal solution and is thus restricted to narrow domains where finding the exact optimal solution is computationally feasible. This naturally led to an extensive body of work on incentive compatible approximation algorithms for domains where the optimal solution is computationally infeasible (see [34] for a survey).

There have been numerous successes with designing incentive compatible algorithms in the era of large scale computation of the commercial internet. Along with the successes, however, there are still domains where, despite extensive research, no *incentive compatible* approximation algorithms with reasonable approximation ratios are known, despite the fact that the problems are easy to approximate in the absence of the incentive compatibility requirement. This led to the strong suspicion that there is an inherent impossibility that prevents the design of computationally feasible and incentive compatible approximation algorithms. This has been the central open problem in algorithmic mechanism design since its inception more than a decade ago [45].

> *Is incentive compatibility inconsistent in general with computationally feasible approximation?*

In the first part of this thesis we settle this long-standing open question. We introduce the *combinatorial public project* model which captures markets that aggregate agents' preferences in networked environments in Chapter 2. From a purely computational perspective, the problem is benign since it has a computationally feasible approximation algorithm that obtains a good (constant factor) solution, assuming that agents report their information truthfully. From a purely strategic perspective the problem is also easy since the VCG mechanism can obtain the optimal solution, assuming it has unlimited computational resources. However, an algorithm with a reasonable approximation ratio that is both incentive compatible and computationally feasible does not exist.

Result. *There is a class of problems that are algorithmically approximable but have no reasonable computationally-feasible incentive compatible approximation.*

The theorem implies that, despite its wide applicability, incentive compatibility can create a severe computational bottleneck for approximation algorithms. In domains where computing optimal solutions can be infeasible, this result shows that incentive compatibility can be an unobtainable goal.

The result is actually made up of two complementary results – one in the communication complexity model and one in the computational complexity model. Both these hardness results heavily rely on a combinatorial characterization of truthful algorithms for our problem, showing that all incentive compatible mechanisms in this class of public projects must be *affine maximizers*. Beyond its implications, our result reveals deep connections between mechanism design, combinatorics and complexity theory and provides a useful and general technique to show computational complexity impossibility results in mechanism design.

Given the impossibility result, in Chapter 3 we investigate the model to identify classes where computationally feasible incentive compatible mechanisms can be obtained. We show computational complexity impossibility results for affine maximizers even in very restricted classes of combinatorial public projects, as well as a computationally feasible mechanism with a good approximation guarantee in a limited setting. We also show several inapproximability result of the computational problem (without incentive constraints) in slightly broader classes. Together, these results provide a map of the computational feasible regions where incentive compatibility may be obtainable in combinatorial public projects.

1.4 Part II: Possibilities of Algorithmic Mechanism Design

Despite the general hardness of incentive compatibility discussed above, it is a powerful solution concept and the golden standard market designers aim for, whenever possible. The bottleneck for incentive compatibility, however, is not necessarily computational. In many emerging online markets the main barrier to implementation comes from the potential overpayments associated with incentive compatibility.

The new generation of the web enables users to easily create and share content as well as receive payments and other forms of incentives in exchange for their online services. This gives birth to new markets that procure information and services from agents and aim to optimize complex objectives under a budget. But there is a clash between budget and incentives: Classical mechanism design theory rules out incentive compatibility under a budget as it is well known that implementing optimal solutions – even for very simple objectives – can result in huge overpayments [7].

Can incentive compatible mechanisms have desirable guarantees under a budget?

In Chapter 4 we introduce the model of *budget feasibility,* where the goal is to provide desirable approximation guarantees to combinatorial objectives while keeping the incentive compatible payments under a fixed budget. The model was originally motivated by the ambition to design incentive compatible mechanisms for the influence maximization problem in social networks. In this setting, first formalized in [22, 33], the goal is to procure a small set of individuals to recommend a technology so that the word-of-mouth effect is maximized. The main result shows that for the rich class of submodular procurement markets near-optimal *approximation* guarantees are obtainable.

Result. *For submodular procurement markets, there are incentive compatible mechanisms that guarantee to provide a constant fraction of the optimal utility without overpayments.*

The main obstacle in this line of work is not computational, though computational thinking helps address the problem in a productive way, through the concept of approximation. This result uses a novel characterization that draws a connection between incentive compatible payments and the approximability of the underlying objective function.

The budget feasibility model is essentially a framework for optimization under incentive constraints. In Chapter 5 we introduce several important techniques for developing budget feasible mechanisms with desirable guarantees in a broad class of problems. In general, the budget feasibility model allows designing powerful mechanisms that can be implemented in real systems, operate efficiently on large scale data and do well in theory and in practice. Beyond the scope of influence maximization and procurement, this framework is useful in data mining. The emergence of crowdsourcing and user-generated content on the web makes incentive-based mechanisms an important addition to the arsenal of data mining techniques.

1.5 Organization of the Thesis

The thesis is divided into two parts. The first part presents the impossibility of algorithmic mechanism design through the combinatorial public projects model. The introduction of the model and the impossibility result are presented in Chapter 2 (based on joint work with Christos Papadimitirou and Michael Schapira [48]), and the results on the mechanisms and approximability of various classes in the model are presented in Chapter 3 (based on work with Michael Schapira [54] and work with Dave Buchfuhrer and Michael Schapira [13]).

The second part of the thesis shows the possibilities of algorithmic mechanism design through the budget feasibility model. The model and main result are presented in Chapter 4 (based on [56]), and further generalizations of approximation techniques are in Chapter 5 (based on joint work with Shahar Dobzinski and Christos Papadimitriou [20] and the work in [57]).

Part I

Limitations of Algorithmic Mechanism Design

Chapter 2

Combinatorial Public Projects I: The Hardness of Being Truthful

2.1 Introduction

The challenge of optimally allocating a *subset* of available resources, that are to be *collectively* shared by many agents, arises in various settings. Consider the following example.

Overlay Networks. In communication networks, routing is done by routers that choose paths other than shortest, and this results in distance matrices that do not satisfy the triangle inequality. In such a situation, it is often desirable to construct an *overlay:* A set of k nodes spread throughout the network and with high-quality routing between them, so that other nodes can improve the distance between them by routing through the closest overlay node. An overlay is beneficial to different nodes in different degrees, and we wish to design the overlay that maximizes total welfare. Beyond the computational challenge in finding the optimal outcome, this problem raises another challenge: To maximize their individual benefit, agents may falsely report their preferences over resources. Therefore, in such settings we are faced with the task of designing a *mechanism* which not only finds (or approximates) the optimal allocation of resources, but also incentivizes the participating agents to be *truthful.*

Combinatorial Public Projects

This is a typical instance of a general problem we define, called COMBINATORIAL PUBLIC PROJECT PROBLEM, or CPP. There are n *agents* and m *resources*, and, for each agent i, a *private valuation function* v_i which specifies i's value for every subset of size k (a parameter) of the resources. We assume that all valuations are *nondecreasing and submodular* (a case that encompasses among many others the overlay network example above, see definitions in

Section 4.2). The objective is to find a subset of the resources $S, |S| = k$, which maximizes the *social welfare*, i.e., the sum of agents' values for the chosen subset $\Sigma_i\ v_i(S)$.

- **The problem is easy from a (strictly) computational perspective.** While many of its special cases are NP-hard [24]), this problem is a submodular maximization problem under a cardinality constraint, and it is well known that a greedy algorithm achieves a constant factor approximation ratio (specifically, $e/(e-1)$) [43]. But the fact that valuations are private presents us with a formidable problem: unless otherwise incentivized, agents are likely to *lie,* exaggerating the degree to which they prefer one alternative over another and this misrepresentation makes optimization impossible.
- **The problem is easy from a (strictly) strategic perspective.** There is a very general method for providing incentives for the agents to reveal their true valuation, namely the *Vickrey-Clarke-Groves (VCG) mechanism* [59, 16, 30]. However, VCG requires that we solve *exactly* (typically many instances of) the CPP — an NP-hard problem. We could of course turn to approximation, as we always do when faced with intractability. The tragedy of this area is that *approximation and truthfulness do not mix:* Running VCG with approximate solutions is, in general, *not* incentive compatible [46].

In other words, efficient approximability and incentive compatibility seem to be at loggerheads. This tension underlies much of the work in algorithmic mechanism design, and is in fact the central problem in the field:

How hard is polynomial-time truthfulness?

In this chapter, we address this long-standing open question and establish a huge gap between the quality of the solutions that can be obtained by algorithms for CPP that are *both* polynomial *and* truthful, and by algorithms that satisfy only *one* of these two desiderata. We show this by proving that *no* truthful *and* computationally-efficient algorithm for CPP can obtain any reasonable approximation ratio, specifically, a ratio better than $\sqrt{m}$ (this holds even for the case of two agents)[1]. This settles a long-standing open question in algorithmic mechanism design, as we exhibit a problem that is easy from a computational perspective (a constant approximation algorithm exists), and from an economic perspective (an optimal truthful algorithm exists), but is hard if we care about both.

Main Result

The main result in this chapter is actually made up of two complementary results – one in the communication-complexity model and one in the computational-complexity model.

[1]In the next chapter we show a simple truthful polynomial-time algorithm that obtains a $\sqrt{m}$ approximation ratio for CPP thus showing that our result is tight in both models.

Theorem. *No polynomial-time truthful mechanism for the combinatorial public projects with submodular agents can obtain an approximation ratio of $O(m^{\frac{1}{2}-\epsilon})$, for every $\epsilon > 0$, unless it uses exponential communication in m. In case of succinctly described valuations, no polynomial-time truthful mechanism can obtain this approximation ratio unless $NP \subseteq BPP$.*

Overview of the Proof

The family of *affine-maximizers* plays a dominant role in the impossibility results. Informally, a range of an algorithm is the collection of all the possible subsets it can output and an algorithm is called an *affine-maximizer* if it *always* optimizes over its entire range (hence, affine maximizers are also known as *maximal-in-range* mechanisms). The proof of the impossibility result consists of two main parts:

1. **Any truthful algorithm is an affine-maximizer (Section 2.3)**. We show that *any* truthful algorithm for CPP is, in fact, an affine maximizer. This part of the proof is inspired by various works that deal with the characterization of truthful mechanisms (see [35, 9, 51, 36, 21]). This line of research was originated in the seminal work of Roberts [50], who has shown that in general domains with unrestricted agents' preferences affine-maximization is necessary for truthfulness. We prove that this is the case even in our restricted setting (in which the agents have normalized, non-decreasing and submodular preferences). The proof shows, for any truthful algorithm, the existence of agent- and outcome-weights as required in the definition of affine maximizers. This is done by exploiting the strong-monotonicity constraint introduced in [35], ideas from the proof of Roberts' Theorem [50], and constructions of non-decreasing submodular functions inspired by those exhibited in [26].

2. **Affine-maximizers cannot achieve good approximations (Sections 2.4 and 2.5)**. We show that, despite the fact that a good approximation exists for the computational problem, no truthful and polynomial-time algorithm can obtain an approximation ratio (asymptotically) better than $\sqrt{m}$. The proof of this result relies on the characterization, and then proving a hardness result for affine-maximizers. We show this in two different models:

   - **Communication-complexity model.** We show that in order to provide a reasonable approximation, any affine maximizer must communicate exponentially many bits. We do this by proving a lower bound on the number of solutions in the range of the mechanism, and then exploiting affine maximization to to establish the communication-complexity lower bound.
   - **Computational-complexity model.** Informally, for the communication complexity lower bound to hold, the agents are assumed to compute their valuations based on exponentially large data. Since many interesting valuations depend on very succinct data, *computational-complexity* techniques would be needed to

establish lower bounds for these. To establish this, we first show, very much as in the proof of our communication-complexity result, an exponential lower bound on the number of solutions, and then we use a probabilistic version of the Sauer-Shelah Lemma [52, 55]. This lemma is closely related to the notion of the Vapnik-Chervonenkis (VC) dimension and has interesting projections on complexity theory.

Organization of the Chapter

After presenting the model formally in Section 4.2, we first show the characterization lemma in Section 2.3. The communication complexity lower bound is then proved in Section 2.4, followed by the computational complexity result in 2.5. We discuss the results and techniques in Section 2.6.

2.2 The Model

In CPP there is a set of n *agents* $\{1, ..., n\}$, and a set of m *resources* $\{1, ..., m\}$. Each agent has a *private valuation function* $v_i : 2^{[m]} \to \mathbb{R}+$. We denote by V_i the space of possible valuations of agent i, and by V the *domain* of valuations $V_1 \times \ldots \times V_n$. We shall assume that $v_i(\emptyset) = 0$. Throughout this chapter, we assume that every v_i is *submodular*:

Definition 1. *A valuation funciton $v : 2^{[m]} \to \mathbb{R}_+$ is* submodular *if for every $S, T \subseteq [m]$ $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$.*

Submodularity is known to be equivalent to the following easily verifiable property, called *decreasing marginal utilities*: For every $S \subset T \subseteq [m]$, and for every $j \in [m]$ such that $j \notin S, T$ $v(S \cup \{j\}) - v(S) \geq v(T \cup \{j\}) - v(T)$. Submodularity arises in many contexts – both economic and computational (see [37, 17, 25] and references therein). Non-decreasing submodular functions are known to have particularly good properties; for example, they can be approximated within a ratio of $e/(e-1)$.

The objective in the CPP is to find a subset of size k, where k is a parameter of the problem, of resources which maximizes the *social-welfare*. That is, we wish to find $T \in \text{argmax}_{S \subseteq [m],\ |S|=k} \Sigma_i v_i(S)$. We are interested in algorithms (*mechanisms*) for CPP satisfying three desiderata:

1. **Quality of solution.** We want our mechanisms to return a solution (set of resources) whose social welfare is as close, in terms of ratio, to the optimum as possible.

2. **Computational Efficiency.** Our algorithms should run in polynomial time. However, since the valuations can be exponentially expressive, one must be careful when defining the input to the problem — the input is the data enabling each agent to compute the valuation. In mechanism design one often takes a "black box" approach: We

assume that valuations are computed by an oracle that can answer a certain type of queries, and we restrict algorithms to ask a polynomial number (in n and m) of such queries. There are two common types of queries:

- In a *value query* the query is a subset of resources $S \subseteq [m]$, and the answer is simply $v_i(S)$; we use this weaker model in our algorithms.
- The *general query* model is equivalent to *Yao's communication model* [61], in which the agents take turns announcing messages; a message by agent i is any function (even a computationally intractable one) of the values of v_i and of the previous messages. We use this stronger model for our impossibility results.

A different approach would be to consider cases in which the "input" (valuations) can be concisely represented, i.e., can be encoded in a natural way that is polynomial in the natural parameters of the problem – m and n. We follow this approach in Section 2.5.

3. **Truthfulness.** That is, we want an algorithm (*mechanism*) A to be such that the agents are rationally motivated to truthfully answer the algorithm's queries. This is achieved by a *payment function* p which, for every n-tuple of valuation functions $v = (v_1, ...v_n) \in V$, demands a payment from each agent. p is such that no agent can increase his/her utility (the value of the set chosen by the algorithm minus the payment assigned to her) by misreporting her valuation[2]. Formally, for every $i \in [n]$ we have that:

$$v_i(A(v_i, v_{-i})) - p(v_i, v_{-i}) \geq v_i(A(v_i', v_{-i})) - p(v_i', v_{-i}), \quad \forall v_i, v_i' \in V_i, \forall v_{-i} \in V_{-i},$$

where V_{-i} is the cartesian product of all V_j's such that $j \neq i$, (v_i, v_{-i}) is the valuations profile in which i has v_i and the other agents have v_{-i}, (v_i', v_{-i}) is defined similarly, and $A(v)$ is the set A outputs for the valuation profile v.

2.3 The Characterization Lemma

Affine Maximizers

In the first step of the proof we restrict our attention to the important family of mechanisms that are *affine-maximizers*. Let A be an algorithm. We define A's *range* R_A to be the resource-subsets of size k that are outputted by A for *some* input, i.e., $R_A = \{S| \ \exists v = (v_1, ..., v_n) \ s.t. \ A(v) = S\}$.

Definition 2.3.1 ([50]). *An algorithm A is said to be an affine maximizer if there exist non-negative agent-weights $w_1, ..., w_n$ (not all equal to 0), and constant outcome-weights $\{C_S\}_{S \in R_A}$ such that*

$$\forall v = (v_1, ..., v_n) \ A(v) \in argmax_{S \in R_A}[(\Sigma_i w_i v_i(S)) + C_S].$$

[2]The notion of truthfulness we consider is the standard notion of truthfulness in dominant strategies. All our results apply to the weaker notion of truthfulness in ex-post Nash.

It is well-known (see, e.g., [50, 45, 35, 19]) that affine-maximizers can be made truthful by enforcing "weighted-VCG" payments [60, 16, 30].

Let A be a truthful algorithm in the combinatorial public projects problem and let R_A be A's range. We wish to show that A is an affine maximizer. The common approach to prove that A is an affine maximizer is to study the topological structure of vectors of *valuation differences*(see e.g. [50, 36]). Let $S \neq T \in R_A$. For any pair of valuation functions $v = (v_1, v_2)$, we shall denote by $v(S) - v(T)$ the two-dimensional vector in $\mathbb{R}^2$ $(v_1(S) - v_1(T), v_2(S) - v_2(T))$. We also define:

$$P(S,T) := \{\alpha \mid \exists\ v\ s.t.\ A(v) = S\ and\ v(S) - v(T) = \alpha\}. \tag{2.1}$$

If A is an affine maximizer that outputs a set S for the valuation functions $v = (v_1, v_2)$ then it must hold that for every $T \neq S$ in R_A (for some *fixed* agent-weights w_1, w_2 and outcome-weights $\{C_R\}_{R \in R_A}$)

$$\Big(\Sigma_i w_i v_i(S)\Big) + C_S \geq \Big(\Sigma_i w_i v_i(T)\Big) + C_T, \tag{2.2}$$

which implies that:

$$\Sigma_i w_i \Big(v_i(S) - v_i(T)\Big) + (C_S - C_T) \geq 0. \tag{2.3}$$

Informally, observe that the inequalities above suggest that if A is an affine maximizer, then there is a line l in $\mathbb{R}^2$ of the form $w_1 x + w_2 y = 0$ such that *every* $P(S,T)$ has l as its lower boundary (possibly shifted by some constant $\gamma(S,T) = C_S - C_T$ from the centre of the axes). So, to prove that a truthful algorithm A is an affine maximizer, we need to show that there are weights w_1, w_2 (and $\{C_R\}_R$) that induce such a line l (the same l for all choices of $S \neq T \in R_A$). This is precisely what we mean to show. The reader is referred to [36] for an explanation of the geometric intuition behind the proof of Roberts' Theorem (which also underlies our proof).

Strong Monotonicity

Strong monotonicity was introduced in [35], and shall be extremely useful to us throughout this proof.

Definition 2.3.2. *An algorithm satisfies strong-monotonicity if for every $i \in [n], v_i, v'_i \in V_i$, and $v_{-i} \in V_{-i}$, if $A(v_i, v_{-i}) = S$ and $A(v'_i, v_{-i}) = T \neq S$ then it must hold that $v_i(S) - v_i(T) > v'_i(S) - v'_i(T)$.*

Not any truthful algorithm is necessarily strongly-monotone (and vice-versa). Yet, we shall prove the theorem for strongly-monotone algorithms. At the end of the proof (see

2.3) we shall revisit this assumption and explain why it can be removed. The following proposition shall play a crucial role in our proofs.

An important element in the proof is the construction of the following function:

$$\forall R \; v(R) = \alpha|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$

Observe that this function has the following properties:

- The function is submodular.

Proposition 2.3.3. *Let A be a strongly monotone algorithm, and let $v \in V$ be such that $A(v) = S$. If $v' \in V$ and $v(S) - v(T) \leq v'(S) - v'(T)$ (i.e., $\leq$ in each coordinate) then $A(v') \neq T$.*

Proof. Assume, for point of contradiction, that the conditions stated in the theorem hold and $A(v') = T$. Let $v = (v_1, v_2)$ and $v' = (v'_1, v'_2)$. Let $\alpha_1 = v_1(S) - v_1(T)$. We shall prove the proposition for the case that $\alpha_1 \geq 0$ (the other case requires a very similar construction). Let $j \in S \setminus T$ (since all sets are of equal size such a j is guaranteed to exist). We define a valuation function $v''_1 \in V_1$ as follows:

$$\forall R \; v''_1(R) = \alpha_1|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$

As discussed in the previous section, this is indeed a non-decreasing submodular function. We shall now show that $A(v''_1, v_2) = S$. This is because if $A(v''_1, v_2) = Q \neq S$ then, by strong monotonicity, $v_1(S) - v_1(Q) > v''_1(S) - v''_1(Q)$. It is easy to show that if $Q \neq S, T$ then this results in a contradiction (as we can set the value of β to be as high as we like). Observe that if we set $Q = T$ then this too results in a contradiction. Similarly, we can show that $A(v''_1, v'_2) = T$. However, in this case by strong monotonicity we have that $v_2(S) - v_2(T) > v'_2(S) - v'_2(T)$. A contradiction. □

Main Part of the Characterization Proof

We shall now prove our result for the case that $\overrightarrow{0} = (0, 0) \in P(S, T)$ for every $S \neq T \in R_A$. This greatly simplifies the exposition and enables us to convey the main idea of the proof. The technical results we shall present here are sufficient to imply the theorem for the more general case using the exact same logic as presented in [36] (first proof in that paper).

Claim 2.3.4. *Let $\alpha \in P(S, T)$ for some $S, T \in R_A$. Let $\epsilon = (\epsilon_1, \epsilon_2) \geq 0$. Then, $\alpha + \epsilon \in P(S, T)$.*

Proof. Since $\alpha = (\alpha_1, \alpha_2) \in P(S, T)$ then, by definition, there are valuation functions $v = (v_1, v_2)$ such that $A(v) = S$ and $v(S) - v(T) = \alpha$. We prove the claim for the case $\alpha \geq 0$ (other cases are handled similarly). Let $j \in S \setminus T$. We define valuation functions $v' = (v'_1, v'_2)$ as follows:

$$\forall R\ v_1'(R) = (\alpha_1 + \epsilon_1)|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$
$$\forall R\ v_2'(R) = (\alpha_2 + \epsilon_2)|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$

The use of strong monotonicity (as in the proof of Proposition 2.3.3) and of Proposition 2.3.3 itself shows that $A(v_1', v_2) = S$. Similarly, we can then show that $A(v_1', v_2') = S$. Observe that $v'(S) - v'(T) = \alpha + \epsilon$. Therefore, by definition, $\alpha + \epsilon \in P(S,T)$. □

Claim 2.3.4 tells us something important about the structure of the different $P(S,T)$ sets in $\mathbb{R}^2$: If a point is in $P(S,T)$ then so are all points "above" it and "to the right" of it. Hence, each $P(S,T)$ is defined by a lower boundary with a non-increasing slope. However, we do not yet know that this non-increasing slope is a straight line (and certainly not that it is the same straight line for all choices of S,T). We shall require the two following technical proposition regarding the correlations between different $P(S,T)$ sets:

Proposition 2.3.5. *For any $S \neq T \in R_A$ $\alpha \in P(S,T)$ iff $-\alpha \notin P(T,S)$.*

Proof. Let $\alpha = (\alpha_1, \alpha_2) \in P(S,T)$. Then, by definition, there exists $v = (v_1, v_2)$ such that $A(v) = S$ and $v(S) - v(T) = \alpha$. Suppose, for point of contradiction, that $-\alpha \in P(T,S)$. Then, by definition, there exists $v' = (v_1', v_2')$ such that $A(v) = T$ and $v'(T) - v'(S) = -\alpha$ (or, $v'(S) - v'(T) = \alpha$). However, by Proposition 2.3.3 this is impossible ($A(v')$ cannot equal T).

We now prove the other direction, which is equivalent to showing that if $\alpha \notin P(S,T)$ then $-\alpha \in P(T,S)$. Since S is in R_A there must be valuation functions $v = (v_1, v_2)$ such that $A(v) = S$. Let $\alpha^W = v(S) - v(W)$. We prove the proposition for the case $\alpha \geq 0$ (other cases are handled similarly). Let $j \in S \setminus T$. We define valuation functions $v' = (v_1', v_2')$ as follows (we choose β to be huge, and in particular higher than the values of all coordinates of all the different α^W's):

$$\forall R\ v_1'(R) = \alpha_1|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$
$$\forall R\ v_2'(R) = \alpha_2|R \cap \{j\}| + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$

The repeated use of Proposition 2.3.3 for every $W \in R_A$ such that $W \neq S,T$ shows that $A(v_1', v_2')$ must be in $\{S,T\}$. However, since $v'(S) - v'(T) = \alpha$ and $\alpha \notin P(S,T)$, it must be that $A(v') = T$. Observe that $v'(T) - v'(S) = -\alpha$. So, by definition of $P(T,S)$, $-\alpha \in P(T,S)$. □

Proposition 2.3.6. *For any $S,T,W \in R_A$, such that no two are equal, if $\alpha \in P(S,T)$ and $\alpha' \in P(T,W)$ then $\alpha + \alpha' \in P(S,W)$.*

Proof. Let $\alpha = (\alpha_1, \alpha_2)$. Let $\alpha' = (\alpha_1', \alpha_2')$. Let Y be an additive valuation function such that $\forall j \notin S \cup T \cup W\ Y(\{j\}) = 0$ and the two following properties hold:

- $\Sigma_{j \in S} Y_j - \Sigma_{j \in T} Y_j = \alpha_1$

- $\Sigma_{j\in T}Y_j - \Sigma_{j\in W}Y_j = \alpha'_1$

Observe that because the number of variables is greater than the number of equations such a function Y exists. Since S, T, W are of equal size we can also assume that Y only assigns non-negative values (otherwise increase the value of each $j \in S \cup T \cup W$ by some identical large enough constant).

Similarly, let Z be an additive valuation function such that $\forall j \notin S \cup T \cup W$ $Z(\{j\}) = 0$ and the two following properties hold:

- $\Sigma_{j\in S}Z_j - \Sigma_{j\in T}Z_j = \alpha_2$
- $\Sigma_{j\in T}Z_j - \Sigma_{j\in W}Z_j = \alpha'_2$

We define (for some huge β to be determined later) the following valuations $v' = (v'_1, v'_2)$:

$$\forall R\ v'_1(R) = Y(R) + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$
$$\forall R\ v'_2(R) = Z(R) + \beta|S||T| - \beta(|S| - |S \cap R|)(|T| - |T \cap R|)$$

Since $\alpha \in P(S,T)$ there must be some valuations $v = (v_1, v_2)$ such that $A(v) = S$. The repeated use of Proposition 2.3.3 (as in the proof of Proposition 2.3.5) shows that $A(v') \in \{S, W\}$. Similarly, by taking advantage of the fact that $\alpha' \in P(T,W)$ one can show (using similar arguments) that $A(v') \in \{S, T\}$. We conclude that $A(v') = S$. Observe that $v'(S) - v'(W) = (v'(S) - v'(T)) + (v'(T) - v'(W)) = \alpha + \alpha'$. Hence, by definition of $P(S,W)$ $\alpha + \alpha' \in P(S,W)$. □

Corollary 2.3.7. *For every $S \neq T, U \neq W \in R_A$ it holds that $P(S,T) = P(U,W)$.*

Proof. Let $\alpha \in P(S,T)$. As $\overrightarrow{0} \in P(T,W)$, we have that $\alpha = \alpha + \overrightarrow{0} \in P(S,W)$. We also know that $\overrightarrow{0} \in P(U,S)$, and so $\alpha = \overrightarrow{0} + \alpha \in P(U,W)$. □

That is, all the $P(S,T)$ sets (for every choice of S,T) are, in fact, the very same set, that we shall refer to as X. We shall now prove that X is convex, thus showing that the (lower) boundary of X (which we know is non-increasing) must be a straight line[3]. Our proof for the convexity of X relies on the assumption that the domain of valuations V is open. Unfortunately, it is *not* true that the domain of nondecreasing submodular valuations is open. At the end of the Characterization Lemma's proof we revisit this assumption and show how it too can be removed.

Definition 2.3.8. *A domain of valuations V is open if for each $v = (v_1, ..., v_n) \in V$, there is some $\epsilon > 0$ such that for all $v' = (v'_1, ..., v'_n)$, if $\forall S \subseteq M$ and $\forall i \in [n]$, $|v'_i(S) - v_i(S)| \leq \epsilon$ then $v' \in V$.*

[3]Observe that if we define $-X = \{-\alpha |\ \alpha \in X\}$ then the convexity of X implies the convexity of $-X$. If X and $-X$ are convex, their union is $\mathbb{R}^2$, and their interiors are disjoint (by Proposition 2.3.5), then they must be separated by a straight line.

Proposition 2.3.9. *X is convex.*

Proof. To prove this proposition we first show that if $\alpha, \alpha' \in X$ then $\frac{\alpha+\alpha'}{2} \in X$. Suppose, by contradiction, that $\alpha, \alpha' \in C$ but $\frac{\alpha+\alpha'}{2} \notin X$. By Proposition 2.3.6 we know that $\alpha + \alpha' \in X$, and by Proposition 2.3.5 we know that $-\frac{\alpha+\alpha'}{2} \in X$. However, Proposition 2.3.6 can then be used to show that $\frac{\alpha+\alpha'}{2} = (\alpha + \alpha') + (-\frac{\alpha+\alpha'}{2}) \in X$. A contradiction.

By repeatedly using this fact, we can, for any $\alpha, \alpha' \in X$ and $\lambda \in (0, 1)$ build a series of points that approach $\lambda\alpha + (1 - \lambda)\alpha'$, such that any point in the series has a ball of small radius that is fully contained in X. This suffices to prove that $\lambda\alpha + (1 - \lambda)\alpha' \in X$. □

Now we know that X is convex and therefore has a lower boundary in the form of a straight line l. Moreover, l goes through the origin $\overrightarrow{0}$, as by Proposition 2.3.5 $\alpha \in C$ iff $-\alpha \notin X$. So, l can be described by $w_1 x + w_2 y = 0$ for some positive constants w_1, w_2 (recall that l's slope is non-increasing). Therefore, we get that A is an affine maximizer (for agent-weights w_1, w_2 and by setting all outcome-weights to be 0). This concludes the proof of the theorem.

Removing the Strong Monotonicity Assumption

The second part of the proof (characterizing truthful algorithms) relied on the assumption that the algorithms in question were strongly monotone. In general, this assumption is not justified. However, we can exploit the following machinery to do away with it:

Theorem 2.3.10 ([35])**.** *If a domain of valuations V is open, then for every truthful algorithm A there exists an algorithm A' that satisfies strong monotonicity such that if A' is an affine maximizer then so is A.*

Theorem 2.3.10 implies the following modus operandi: We shall focus on an open subdomain of the domain of all non-decreasing submodular valuation functions. We shall show that all our arguments actually apply to that domain. Since in that domain truthfulness is equivalent to strong monotonicity (in the sense stated by Theorem 2.3.10) this will conclude the proof. We now provide a way to slightly tweak all constructions of valuation functions in our proof to ensure that they all belong to an open domain.

We define the *strict domain* V^{strict} to be the domain of all normalized valuations that are strictly non-decreasing (i.e., $\forall S \subset T \subseteq [m]$ and $\forall i \in [n]$ $v_i(S) < v_i(T)$) and have strictly decreasing marginal utilities. (i.e., $\forall S \subset T \subseteq [m]$, $\forall j \notin S, T$ and $i \in [n]$ $v_i(S \cup \{j\}) - v_i(S) > v_i(T \cup \{j\}) - v_i(T)$). The strict domain is open: Consider some $v = (v_1, ..., v_n) \in V^{strict}$. Let δ be the minimal gap caused by the above strict inequalities (taken over all possible sets of resources, agents, etc.). It is easy to see that $\epsilon = \frac{\delta}{3}$ meets the requirement in the definition of an open domain.

We now explain how to ensure that a profile of valuations $v \in V$ is in V^{strict}. Let ϵ' be some positive number. Let $g_{\epsilon'}$ be a valuation function such that $\forall R \subseteq [m]$ $g_{\epsilon'}(R) = |R|\epsilon' - \frac{2^{|R|}}{2^{m+1}}\epsilon'$. The reader can verify that this function is strictly non-decreasing and has strictly decreasing marginal utilities. Moreover, for any non-decreasing submodular valuation function and for any ϵ' it holds that $v_i + g_{\epsilon'}$ is *strictly* non-decreasing and has *strictly* decreasing marginal utilities. Hence, we can convert all the valuation functions in the proof to valuations that uphold the required properties using this gadget. The reader may verify that (for sufficiently small choices ϵ') all the arguments in this proof hold for these slightly different constructions of valuation functions.

2.4 Communication Complexity Lower Bound

The communication complexity lower bound is proved in two steps:

- Informally, we start by showing, via the probabilistic method, that *any* algorithm (not necessarily an affine maximizer) that obtains an approximation ratio better than $\sqrt{m}$ must have a "huge" range (exponential in m). (This is, in fact, true even for $n = 1$.)
- We then show, via a communication complexity reduction, that the fact that affine maximizers must optimize over the *entire* exponential-sized range necessitates the transmission of exponentially many bits by the algorithm. (This is true even for $n = 2$.)

We set k to be $\sqrt{m}$ (throughout the proof). Let A be an algorithm, and let R_A be A's range. In our proof we will use *additive* valuations which are a special case of submodular valuation functions.

Definition 2.4.1. *A valuations function $v : 2^{[m]} \to \mathbb{R}_+$ is called* additive *if there exist weights $w_1, \ldots, w_m$ where $w_i \in \mathbb{R}_+ \ \forall i \in [m]$ and $v(S) = \sum_{i \in S} w_i$.*

Lemma 1. *For any algorithm A that obtains an approximation ratio of $m^{\frac{1}{2}-\epsilon}$ to the optimal social welfare it must hold that $R_A = \Omega(e^{m^\epsilon})$. This is true even for $n = 1$ and even if the single agent has an additive valuation function.*

Proof. Let A be an algorithm as in the theorem statement. We shall prove the theorem for the case that there is 1 agent with an additive valuation function. We will use a probabilistic construction to show that obtaining an approximation ratio better than $m^{\frac{1}{2}}$ requires the range of allocations to be exponentially large in m. Let $\epsilon > 0$. First, we construct a set T by choosing each resource in M to be in T with probability $m^{-\frac{1}{2}+\epsilon}$ (an independent random experiment for each resource). Let v_T be the *additive* (and thus submodular) valuation function defined in the following manner:

$$v_T(j) = \begin{cases} 1 & j \in T \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

Observe that for any $S \subseteq [m]$, $v_T(S) = |S \cap T|$. We would like to show that the probability that the value $v_T(S)$ of a set $S \in R_A$ approximates the optimal social welfare within a ratio of at least $\sqrt{m}$ is exponentially small in m. In our construction this is equivalent to showing that the probability of $|S \cap T|$ being larger than m^ϵ and the probability of $|T|$ being smaller than $m^{\frac{1}{2}+\epsilon}$ are exponentially small. We will show these two claims separately using the Chernoff bound.

Claim 2.4.2. *(Chernoff Bound) Let $X_1, \ldots, X_t$ be a set of t independent random variables that take values in $\{0, 1\}$ such that for every i, $Pr[X_i = 1] = p$. Then, for any δ is in the range $[0, 2e - 1]$ we have that:*

$$Pr[\sum_{i=1}^{t} X_i > (1 + \delta)pt] \le e^{\frac{-\delta^2 pt}{3}} \tag{2.5}$$

$$Pr[\sum_{i=1}^{t} X_i < (1 - \delta)pt] \le e^{\frac{-\delta^2 pt}{3}} \tag{2.6}$$

Fix some $S \in R_A$. For each resource $i \in S$, let X_i be the following random variable:

$$X_i = \begin{cases} 1 & i \in T \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

Due to our construction we have that $Pr[X_i = 1] = m^{-\frac{1}{2}+\epsilon}$ for every $i \in \{1, \ldots, \sqrt{m}\}$. By 2.5 we have:

$$Pr[|S \cap T| > (1 + \delta)m^\epsilon] \le e^{\frac{-\delta^2 m^\epsilon}{3}}$$

.

Similarly, for the same random variables as above now defined for each resource in $[m]$, using 2.6 to evaluate the number of elements in T we get:

$$Pr[|T| < (1 - \delta)m^{\frac{1}{2}+\epsilon}] \le e^{\frac{-\delta^2 m^{\frac{1}{2}+\epsilon}}{3}}.$$

We can thus infer that the probability of T being "small" or $S \cap T$ being "big" is exponentially small:

$$Pr[|T| < (1 - \delta)m^{\frac{1}{2}+\epsilon} \text{ or } |S \cap T| > (1 + \delta)m^\epsilon] \le 2 \cdot e^{\frac{-\delta^2 m^\epsilon}{3}}.$$

Using the union bound, it is thus evident that if R_A must include at least $\Omega(e^{m^\epsilon})$ different sets, otherwise there is some additive valuation function v_T such that A fails to attain an approximation of $m^{\frac{1}{2}-\epsilon}$ for a. □

So, any algorithm A (not necessarily an affine maximizer) that obtains an approximation ratio better than $\sqrt{m}$ has a range of size exponential in m. For affine maximizers this fact leads to an inapproximability result.

Lemma 2.4.3. *Any affine maximizer A with range R_A requires $\Omega(|R_A|)$ communication, even for $n = 2$.*

Proof. Consider an affine maximizer A with range R_A. We shall construct two submodular valuation functions which require $\Omega(|R_A|)$ communication in order to achieve optimality in the range R_A.

The proof is by reduction from the Set Disjointness [61]. In this problem there are 2 parties, and each party i holds a string $s_i \in \{0,1\}^d$, and the goal is to tell whether there exists a single index j where $s_1(j) = s_2(j)$ (where $s_i(j)$ denotes the bit of string i at index j). It is known that deciding this problem requires communicating $\Omega(d)$ many bits between the parties.

Recall that A maximizes $w_1 v_1(S) + w_2 v_2(S) + C_S$ over all $S \in R_A$; it is easy to see that one can assume $w_1 = w_2 = 1$ for all S, w.l.o.g. Now, suppose that each agent i has a private set $R_A^i \subseteq R_A$, which induces the following valuation function:

$$v_i(T) = \beta \cdot \left(\Pi_{S \in R_A^i} |S| - \Pi_{S \in R_A^i} (|S| - |S \cap T|) \right) \quad \forall T \subseteq [m]. \tag{2.8}$$

Note that since β can be arbitrarily large, w.l.o.g. we can consider the case in which $C_S = 0$ for all $S \in R_A$. Observe that if A maximizes over the range R_A, it implicitly distinguishes between the case for which R_A^1 and R_A^2 intersect, and the case in which $R_A^1 \cap R_A^2 = \emptyset$. Therefore this is a reduction from the set-disjointness problem, establishing that $\Omega(|R_A|)$ communication is required. □

2.5 Computational Complexity Lower Bound

We now describe a novel technique for deriving *computational-complexity* (*not* communication-complexity) lower bounds for mechanism design problems. In particular, we show that, even if agents have succinctly described valuations, CPP is inapproximable *in polynomial time* by truthful algorithms, unless $NP \subseteq BPP$. Specifically, we prove the following theorem:

Theorem 2.5.1. *There is a class of succinctly-described submodular valuation functions for which no truthful and polynomial-time algorithm cannot achieve an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$ unless $NP \subseteq BPP$ (for every $\epsilon > 0$).*

A Lower Bound for Affine Maximizers

We shall start by proving our lower bound for affine maximizers. We shall then show how this result can be extended to any truthful algorithm. The main part of this proof uses the class of coverage valuation functions, defined as follows.

Definition 2. *A valuation function $v : 2^{[m]} \to \mathbb{R}_+$ is called coverage if there exists subsets $T_1, \ldots, T_m$ of some universe $\mathcal{U}$ where $v(S) = |\cup_{i \in S} T_i|$.*

Coverage functions can be succinctly represented in the size of the universe $\mathcal{U}$ and it is easy to verify they are indeed submodular. This is an important class of submodular functions, since such functions are both combinatorially rich and succinctly representable at the same time. Coverage functions will play an important role throughout this thesis. But first, we take Manhattan.

Lemma 2.5.2. *No polynomial-time affine maximizer for CPP with coverage valuations achieves an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$ unless NP $\subseteq$ BPP (for every $\epsilon > 0$ and even for $n = 1$).*

Proof. Fix $k = \sqrt{m}$ and $n = 1$. Let A be an affine maximizer as in the statement of the theorem (w.l.o.g. assume that the agent-weights are 1 and outcome-weights are 0). R_A consists of subsets of $[m]$ of size $\sqrt{m}$ that are assigned by A to different inputs. Exactly as in Lemma 1, it can be shown that R_A must contain $\Omega(e^{m^\epsilon})$ sets. We now recall the Sauer-Shelah Lemma:

Lemma 2.5.3. *([52, 55]) For any family Z of subsets of a universe R, there is a subset Q of R of size $\Theta(\frac{\log|Z|}{\log|R|})$ such that for each $Q' \subseteq Q$ there is a $Z' \in Z$ such that $Q' = Z' \cap R$.*

Hence, coming back to the affine maximizer, there is a subset M' of $[m]$ of size $\Theta(\frac{m^\epsilon}{\log m})$ that is "shattered" by R_A. The idea is now to embed a small, but still polynomially large, instance of an NP-complete problem in M'.

We shall do this for the NP-complete t-Cover problem where we are given a family F of sets $T_1, \dots, T_\ell$ of some universe $\mathcal{U}$ and the goal is to decide whether there exists a family of subsets of size t that covers $\mathcal{U}$. The embedding is as follows: Given an instance to t-Cover we construct a new universe $\mathcal{U}'$ of size $2|\mathcal{U}|+m-|M'|$. Each item $a \in \mathcal{U}$ will have two copies in $\mathcal{U}'$, and both these copies will be covered by the set S_i for all $i \in \{1, \dots, |M'|\}$ if and only if they are covered by the set T_i in the t-Cover instance. For the rest of the items in $\mathcal{U}'$, each item j will be covered by a single set S_j and a_j is the only item that S_j covers. The coverage valuation over the resources is simply $v(S) = |\cup_{i\in S} S_i|$.

Now observe that the optimal social welfare for v is $2|\mathcal{U}|+k-r$ if and only if there exists a cover of size r of $\mathcal{U}$. If there exists a cover S' of t sets $\{i \in S' : T_i\}$, then these sets contribute $2|\mathcal{U}|$ to the welfare using $\{i \in S' : S_i, \dots, S_r\}$ with any arbitrary set of $k-r$ resources from $[m] \setminus M'$. Conversely, suppose that the optimal social welfare is at least $2|\mathcal{U}|+k-r$ achieved by selecting the subset of resources S, but there exists a cover S' of $\mathcal{U}$ s.t. $|S'| < r$. From our construction this implies that $|\cup_{i\in S'} S_i| = 2|\mathcal{U}|$ and that there $k-r'$ additional resources that can be added from $[m] \setminus M'$, each adding value of 1, and thus the optimal welfare is $2|\mathcal{U}|+k-r' > v(S)$, which is a contradiction to the optimality of S.

The above suggests a *non-uniform* reduction from an NP-hard problem to the problem of calculating the output of the affine maximizer in question. The reason the reduction is non-uniform (and thus it does not establish NP-completeness) is because *we do not know how to find M'* (see [49, 53, 40] on the complexity of making Sauer-Shelah Lemma constructive). However, this non-uniform reduction is sufficient to show that if A obtains an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$ in polynomial time then *NP has polynomial-size circuits*, i.e., NP $\subseteq$ P/*poly*.

By using the probabilistic version of the Sauer-Shelah Lemma presented by Ajtai [3] we can turn the reduction described above into a *probabilistic* polynomial-time reduction (thus concluding the proof of the theorem).

Lemma 2.5.4. *([3]) Let Z be a family of subsets of a universe R that is regular (i.e., all subsets in Z are of equal size) and $Q \geq 2^{|R|^\alpha}$ (for some $0 < \alpha \leq 1$). There are integers q, l (where $|R|, q$ and l are polynomially related) such that if we randomly choose q pairwise-disjoint subsets of R, $Q_1, ..., Q_q$, each of size l, then, w.h.p., for every function $f : [q] \to \{0,1\}$ there is a subset $Z' \in Z$ for which $|Z' \cap Q_j| = f(j)$ for all $j \in [q]$.*

The probabilistic polynomial-time reduction from our NP-complete problem is very similar to the non-uniform reduction shown above. The key idea is finding pairwise-disjoint subsets of $[m]$ as in the statement of Lemma 2.5.4, and then associating each subset of the universe in the NP-complete problem with *all* elements in one of the pairwise-disjoint subsets. □

Extending the Lower Bound to All Truthful Algorithms

We extend our lower bound to all truthful algorithms by relying on the following simple observation: All the submodular functions that are constructed as part of the proof of the Characterization Lemma are, in fact, succinctly described. We can therefore define a class of succinctly described valuation functions C, that contains the families of functions constructed in the Characterization Lemma *and* all coverage valuations. Because of the way we defined C one can show that any truthful algorithm for the CPP with valuations in C is an affine-maximizer. We can now use Lemma 2.5.2 to conclude the proof of the Theorem.

2.6 Discussion

The results in this chapter show that hardness of incentive compatible computation. Although combinatorial public projects have good approximations from a strictly computational perspective, no truthful computationally-feasible mechanism can obtain a reasonable approximation ratio.

An interesting way to view our computational-complexity result is the following: Over the past four decades, complexity theory has been successful in classifying optimization problems into various classes, such as P, NP, NP-hard, and APX (those problems that can be approximated within some constant factor in polynomial time). Mechanism design is about *incentive-compatible optimization*, in which the inputs are provided by agents who have their own objectives. In this new regime, for social-welfare maximization problems classical VCG theory implies that P, NP, and NP-hardness are preserved under truthfulness. *Our result essentially states that APX is* not *preserved.*

The VC dimension technique can be applied to other domains. In the next chapter we exploit this technique to show computational complexity lower bounds for stricter valuation classes. It also has an interesting projection in complexity theory: Call a language $L \subseteq \{0,1\}^*$ *exponentially dense* if there is some $\alpha > 0$, and some integer N, such that for any integer $n > N$ it holds that $|L \cap \{0,1\}^n| \geq 2^{n^\alpha}$. For a language $L \subseteq \{0,1\}^*$, define SAT_L to be the problem: "Given a CNF, is there a truth assignment *in* L that satisfies it?" The proof technique implies that:

Theorem 2.6.1. *Let L be any exponentially dense language. If SAT_L is in P, then NP $\subseteq$ P/poly.*

Observe that unlike CPP we do not know how to relax the computational hardness assumption to NP $\subseteq$ BPP (e.g., via the probabilistic version of the Sauer-Shelah Lemma). For the problem CIRCUIT SAT, however, we can prove this stronger result via a different technique:

Theorem 2.6.2. *Let L be any exponentially dense language. If CIRCUIT SAT_L is in P, then NP $\subseteq$ BPP.*

Proof. (Sketch) To solve a given CNF ϕ on n variables, start with a large enough N so that L contains at least 2^{n^2} strings of length N. Now *hash* these N bits (by a circuit computing a sampled universal hashing function) into n bits. With very high probability, the 2^{n^2} bitstrings in L of length N will cover, after hashing, all 2^n bitstrings of length n. Then feed these n bits into a verifier circuit for ϕ. It is not hard to see that, with high probability, this overall circuit, with N inputs, has a satisfying truth assignment in L if and only if ϕ is satisfiable. $\square$

Chapter 3

Combinatorial Public Projects II: Adventures in Incentives and Computation

3.1 Introduction

The hardness of incentive compatibility presented in the previous chapter showed that in combinatorial public projects with *submodular* agents, despite being a benign computational problem, no polynomial-time truthful mechanism can obtain a reasonable approximation (under standard computational assumptions). The impossibility is due to two fundamental results: for *submodular* valuations *Maximal-In-Range (MIR)* mechanisms (*affine maximizers*) provide poor approximations, and all truthful mechanisms are affine maximizers. Are there cases of combinatorial public projects that can avoid this impossibility result? Specifically, if we consider agents with stricter valuations, could we find truthful polynomial-time mechanisms? Conversely, if we relax the submodularity assumption does truthfulness continue to be a computational burden?

Since public projects is a social welfare maximization problem, the VCG mechanism provides an incentive compatible solution. To apply VCG however, the optimization problem must be computable in polynomial time. Therefore, to find cases where incentive compatible mechanisms can be obtained, the first step is identifying the cases where the optimal solution for CPP can be computed in polynomial time. For the classes of submodular valuations that are NP-hard but have stricter structures, the first question is whether MIR mechanisms provide reasonable approximations. The second question for such valuations is whether it still holds that all truthful mechanisms are necessarily MIR. Since these are the two elements responsible for the impossibility result for submodular valuations, obtaining a truthful mechanism with a reasonable approximation ratio requires circumventing at least one of these barriers.

For classes of valuations that contain submodularity, the impossibility result naturally applies. The question is whether incentive compatibility remains a computational burden. That is, without incentive constraints, does the public projects problem with valuations broader than submodular remain *approximable*?

In this chapter we consider cases where agents' valuation functions are *complement-free*, *i.e.*, cases in which agents' valuations are *subadditive* set functions. We focus on this class since there is no hope in approximating the problem without this assumption as we show in Section 3.5. Hence, due to computational reasons, the class of subadditive valuations is the only general class where we can hope to find interesting special cases of CPP. This class includes the submodular class of valuations for which our impossibility result was shown, and encapsulates a rich family of valuation functions [37, 44]. We will map the computationally-feasible regions in the combinatorial public projects problem. We will find the tractable domains of the problem where polynomial-time mechanisms can be obtained via VCG, as well as the inapproximable ones in which – due to computational limitations – no reasonable approximation guarantees are obtainable. We will also show lower bounds on MIR mechanisms, as well as a non-MIR truthful mechanism in an NP-hard case which obtains a constant factor approximation.

Overview of the Results

We now briefly survey our main results and their implications:

- **Tractability.** For CPP with n agents, we show that even for the lowest (most restricted) class of valuations in the complement-free hierarchy, finding an *optimal* outcome is NP-hard. Specifically, CPP is hard even for *unit-demand* valuations, in which every agent is only interested in getting a single resource. Moreover, going up just one step higher in the hierarchy to capped-additive valuations, CPP becomes hard even for a constant number of agents.

- **Approximability.** Our main inapproximability result is for *fractionally-subadditive* valuations. We show that, unlike the case of CPP with submodular valuations, for fractionally-subadditive valuations, no constant approximation ratio is achievable. We show this both in the communication complexity model and the computational complexity model, where agents have succinct fractionally subadditive valuations. We show an upper which implies these results are nearly tight. We present many other positive and negative approximability results: We also show that the $e/(e-1)$ approximation ratio for CPP with submodular valuations is tight even for unit-demand valuations. By contrast, we present improved ratios for other well-studied subclasses of submodular valuations.

- **Truthfulness.** We present both truthful mechanisms and hardness results for truthful computation. For the class of subadditive valuations, we show a truthful $\sqrt{m}$-approximation mechanism, which is an Affine Maximizer. Since submodular valuations are a strict subset of subadditive, this result implies that our impossibility result from the previous chapter is tight. We also present several inapproximability results for the class of *Maximal-In-Range* (MIR) truthful mechanisms. In particular, we show that no constant approximation ratio is achievable for such mechanisms even with unit-demand valuations. Interestingly, we show a truthful constant-factor approximation for CPP with unit-demand agents, thus establishing a gap between VCG-based and general truthful mechanisms. In the previous chapter we showed that when agents valuations are submodular then MIR are the only truthful mechanisms in combinatorial public projects. The question is whether this is true for stricter classes of combinatorial public projects. We show that for a (very) restricted class we call $2-\{0,1\}$-unit-demand valuations, there are truthful mechanisms that are not MIR that obtain a constant-factor approximation. In addition, we show that no MIR mechanism can obtain an approximation within a factor better than $\sqrt{m}$.

Organization of the Chapter

Each of the sections focuses on exactly one class in the complement-free tree. In Section 3.2 we present our results for unit-demand valuations. Section 3.3, deals with capped additive valuations, and Section 5.6 discusses fractionally-subadditive valuations, respectively. We show impossibility results for general valuations in Section 3.5 and conclude with a brief discussion in Section 3.6.

3.2 Unit-Demand Valuations

The simple class of *unit-demand* valuations, in which every agent is only interested in getting a single resource, constitutes the lowest level of the complement-free tree.

Definition 3. *A function $v : 2^{[m]} \to \mathbb{R}_+$ is called unit-demand if $v(S) = \max_{i \in S} v(i)$, for every $S \subseteq [m]$. Such a valuation is represented by a list of the m values $v(j)$, $\forall j \in [m]$.*

Note that for a constant number of agents the optimization problem is solvable in polynomial time. To see this, let c be the number of agents, and k the number of resources selected in CPP. If $c \leq k$, for each agent we choose the resources she values most. If $c > k$, we simply enumerate over all possible subsets in polynomial time. This immediately implies the following.

Observation 3.2.1. *For a constant number of unit-demand agents there is a truthful polynomial-time mechanism (the VCG mechanism) that obtains the optimal solution.*

Although the problem is solvable in polynomial time when we have a constant number of agents, once we allow for a linear number of agents CPP with unit-demand agents becomes hard. This is rather surprising given the simple structure of unit-demand valuations. It is, however, true.

Theorem 3.2.1. *CPP with n agents that have unit-demand valuations is NP-hard to solve optimally. Furthermore, no algorithm for CPP with n unit-demand valuations has an approximation ratio of $\frac{e}{e-1} - \epsilon$ unless P=NP (for any constant $\epsilon > 0$).*

Proof. We will show an approximation preserving reduction from Max-t-Cover. Recall that in Max-t-Cover there is a collection of subsets $\mathcal{F}$ of a universe $\mathcal{U}$ and an integer t, and the goal is to find t sets in $\mathcal{F}$ which have a union of maximum cardinality. It was shown in [24] that the problem cannot be approximated to within $e/(e-1) - \epsilon$ for any constant $\epsilon > 0$ unless $P = NP$.

Consider a Max-t-Cover instance over set $\mathcal{U}$ with $\mathcal{F} = \{T_1, \ldots, T_\ell\}$ and number of sets to be chosen t. We create a CPP instance with resource set $M = \mathcal{F}$ and $|\mathcal{U}|$ agents, one corresponding to each element of $\mathcal{U}$. The agent corresponding to element i values each item $j \in M$ as:

$$v_i(j) = \begin{cases} 1, & i \in T_j \\ 0, & \text{otherwise} \end{cases}$$

So the value for agent i is 1 if i is covered by the chosen set and 0 otherwise. Thus, the social welfare is the number of covered items, or the cardinality of the union of the chosen sets. By setting the number of resources allowed to be chosen to $k = t$, we see that if we can approximate the social welfare to within any factor α, we get an α-approximation of Max-t-Cover as well. So by [24], an approximation of $e/(e-1) - \epsilon$ is not achievable. $\square$

Observe that the above hardness of approximation result is tight (a simple greedy algorithm obtains an approximation ratio of exactly $e/(e-1)$).

MIR Mechanisms

We next consider the class of *maximal-in-range* (MIR), or *VCG-based* mechanisms. For the rest of this section, our results shall be proven for an even more restrictive class of valuations.

Definition 4. *A valuation function $v : 2^{[m]} \to \{0,1\}$ is called c-{0,1}-unit-demand if it is unit-demand and there are at most c items $j \in [m]$ for which $v(j) = 1$.*

We will show that there are truthful mechanisms that are not MIR that obtain a constant-factor approximation for the class 2-{0,1}-unit-demand (in fact our claims hold for any c-{0,1} when c is a constant). In addition, we show that no MIR mechanism can obtain an approximation within a factor better than $\sqrt{m}$. In Section 5.6, a computationally-efficient MIR mechanism for CPP with subadditive valuations with an approximation ratio of $\sqrt{m}$ is presented. As we now show, this approximation ratio is tight for MIR mechanisms even when restricted to 2-{0,1}-unit-demand valuations.

Theorem 3.2.2. *No computationally-efficient MIR mechanism can approximate CPP with n agents that have 2-{0,1}-unit-demand valuations within $m^{-(\frac{1}{2}-\epsilon)}$ (for any constant $\epsilon > 0$) unless $NP \subset P/poly$.*

Proof. We begin by noting that in Lemma 1 in the previous chapter, it was shown that any algorithm for CPP which achieves an approximation ratio of at least $m^{1/2-\epsilon}$ has a range of size $\Omega(e^{m^\epsilon})$. This proof required that for any $T \subseteq [m]$, it is possible to create a set of agents such that the social welfare is $v_T(S) = |T \cap S|$, and was done with a single additive agent. This is easy to do with n 2-{0,1}-unit-demand agents: For each resource in T we construct an agent that has value 1 for that resource and 0 for all other items. This gives the following useful lemma:

Lemma 2. *Any maximal-in-range mechanism for CPPP with n 2-{0,1}-unit-demand agents which achieves an approximation ratio of at least $m^{1/2-\epsilon}$ must have a range of size $\Omega(e^{m^\epsilon})$.*

From this, we can use the Sauer-Shelah lemma to see that the range has a VC dimension at least m^α for some constant $\alpha > 0$. This large range allows us to perform reductions similar to the ones we use in our NP-hardness proofs to show inapproximability. We begin with the modified unit-demand reduction.

As shown above, any maximal-in-range mechanism which approximates better than $m^{1/2-\epsilon}$ must have a range with VC-dimension at least m^α. Reorder the items such that the m^α corresponding to this VC-dimension are the set $[m^\alpha]$. We show a reduction from Vertex Cover. Let m^α be the number of vertices in the Vertex Cover instance, ℓ be the number of edges and $k' < k$ be the target size of the vertex cover. We will construct an instance to CPP with $2\ell + m - m^\alpha$ agents with 2-{0,1}-unit demand agents, and the number of resources to be selected will be $k > m^\alpha$.

The set of resource will be composed of two sets M_1 and M_2, where every resource in M_1 corresponds to a vertex from Vertex Cover and M_2 is a set of resources of size $m - m^\alpha$. For each edge e_i in Vertex Cover we construct two agents i and $i + \ell$ s.t.:

$$v_i(S) = v_{i+\ell}(S) = \begin{cases} 1, & \exists j \in S \cap e_i \\ 0, & \text{otherwise} \end{cases}$$

Since each edge only includes 2 vertices, each such agent in $[2\ell]$ indeed has a 2-{0,1}-unit-demand valuation. The other set of agents is of size $m - m^\alpha$, and every agent is associated with a single resource in M_2 s.t.:

$$v_i(S) = \begin{cases} 1, & i \in S \\ 0, & \text{otherwise} \end{cases}$$

If a single edge is unsatisfied in Vertex Cover, more social welfare can be obtained by adding a resource from M_1, since a resource in M_1 adds at least 2 to the social welfare where a resource from M_2 only contributes 1 to the social welfare. So if the minimum vertex cover has size k', the maximum social welfare is $2\ell + (k - k')$. Furthermore, the mechanism will find this maximum, as it's range includes every subset of M_1, padded out with arbitrary elements from M_2 to reach size k. Thus, the mechanism can be used to find the size of the minimum vertex cover and therefore cannot run in polynomial time unless $NP \subset P/poly$. □

A (non-MIR) Truthful Mechanism

Theorem 3.2.2 shows that no constant-approximation MIR mechanisms exist even for CPP with 2-{0,1}-unit-demand valuations. In contrast, a simple non-addaptive greedy algorithm achieves a two-approximation for 2-{0,1}-unit-demand valuations that is truthful without payments, thus establishing a large gap between what is achievable via MIR and general truthful mechanisms.

Theorem 3.2.3. *There exists a computationally-efficient and truthful mechanism for CPP with 2-{0,1}-unit-demand valuations that has an approximation ratio of two.*

Proof. Consider the following mechanism:

A Deterministic Mechanism for 2-{0,1}-Unit-Demand

1. For each resource j let $s_j = |\{i : v_i(\{j\}) = 1\}|$.
2. Sort the m resources in decreasing order by the value of s_j, breaking ties in favor of resources with lower indices.

Output: Choose the set S consisting of the k first resources in the above ordering.

The mechanism allows agents to vote for two resources, then chooses the k with the most votes. An agent only benefits from adding votes to the two resources that she actually desires, as adding other items to the top k does not improve her utility. As the two resources are desired equally, there is no advantage in voting for one of the resources the agent desires and not the other. So there is never an incentive for an agent to misreport her valuation.

We will now see that this has an approximation ratio of two. Every agent has a value of either 0 or 1 for the chosen set. If an agent has a value of 1, we call that agent satisfied. For each resource j, let s_j be the number of agents satisfied by j. For any set T, $\sum_{j\in T} s_j$ is an upper bound on the social welfare of T. Clearly, S maximizes $\sum_{j\in S} s_j$ for sets of size k, so $\sum_{j\in S} s_j$ is an upper bound on the maximum social welfare. Furthermore, each agent is satisfied by at most 2 items in S, so the social welfare of S is at least $\sum_{j\in S} s_j/2 = 1/2 \sum_{j\in S} s_j$, which is at least 1/2 the maximum social welfare. □

3.3 Capped Additive Valuations

Intuitively, a capped additive valuation is a valuation function that is additive but cannot exceed some threshold. Recall that an additive valuation is a function where the value for each bundle of resources is the additive sum of the per-resource values.

Definition 5. *A valuation $v : 2^m \to \mathbb{R}_+$ is a* capped additive valuation *if there exists an additive valuation f, and a real value $C > 0$, such that, for each $S \subseteq [m]$, $v(S) = \min\{f(S), C\}$.*

NP-hardness and a FPTAS

Since 2-{0,1}-unit-demand valuations are a subclass of capped additive valuations (where $C = 1$), our negative results in Section 3.2 for CPP with n agents extend to capped additive valuations. What about a constant number of agents? Observe that finding the optimal outcome for a single agent is trivially in P (simply take the k most valued resources). It turns out, however, that even with two agents, the problem is hard.

Theorem 3.3.1. *CPP with* 2 *capped additive valuations is NP-hard.*

Proof. We reduce from Subset Sum, where we are given a set of positive integers $w_1, \dots, w_\ell$ and a target t, and the goal is to find a subset of $\{w_1, \dots, w_\ell\}$ that sums to t. Given an instance of Subset Sum, we construct an instance to our problem with $m = 2\ell$ resources, $k = \ell$, and 2 agents with valuations $v_1(S) = \min\{f_1(S), C_1\}$ and $v_2(S) = \min\{f_2(S), C_2\}$, where $C_1 = 2t$, $C_2 = \ell \cdot \max_j w_j$ and f_1, f_2 are the additive functions defined as follows:

$$f_1(j) = \begin{cases} 2w_j, & j \le \ell \\ 0, & \text{otherwise} \end{cases}$$

$$f_2(j) = \begin{cases} C_2/k - w_j, & j \le m \\ C_2/k, & \text{otherwise} \end{cases}$$

Observe that if there exists a subset S s.t. $\sum_{j \in S} w_j = t$ in Subset Sum, by choosing the set of resources $S' = S \cup \{\ell + 1 \dots 2\ell - |S|\}$ we have social welfare of $v_1(S') + v_2(S') = f_1(S') + f_2(S') = C_2 + t$ in CPP.

Conversely, consider a subset S' of resources in CPP of size $k = \ell$ with social welfare of at least $C_2 + t$. Let $S \subseteq S'$ be the subset of resources with index $i \le \ell$. We will show that the $\sum_{j \in S} w_j = t$ in Subset Sum. Assume, for purpose of contradiction that $\sum_{i \in S} w_j > t$. In this case: $v_2(S) < C_2 - t$ and $v_1(S) = 2t$, thus since $S \subseteq S'$ we have that $v_1(S') + v_2(S') < C_2 + t$ and we get a contradiction. Assume now, again for purpose of contradiction, that $\sum_{i \in S} w_j < t$. Then again $v_1(S') + v_2(S') < C_2 + t$ and a contradiction. It therefore follows that $\sum_{i \in S} w_j = t$ as required. □

Although the problem is hard for two agents, using dynamic programming we can obtain a Fully Polynomial Time Approximation Scheme (FPTAS), for any constant number of agents.

Theorem 3.3.2. *There exists a FPTAS for CPP with a constant number of capped additive valuations.*

Proof. We will use a dynamic programming procedure. Let $b = \max_{i \in n}\{\max_{S:|S|=k} v_i(S)\}$. We divide the interval $[0, b]$ into $\frac{n \cdot m}{\epsilon}$ segments, each of length $\frac{\epsilon b}{n \cdot m}$, and denote $p(x) = \lfloor x \cdot mn/\epsilon b \rfloor$. We will maintain an n-dimensional table with $(\frac{n \cdot m}{\epsilon})^n$ entries, denoted A, where in each entry $A_{ij \dots k}$ we will store a subset S for which $p(v_n(S)) = k, p(v_2(S)) = j, \dots p(v_n(S)) =$

k, if such a subset exists. For convenience, for a given subset S we will denote $A(S)$ to be its corresponding entry in the table.

Assume some arbitrary ordering $\{1 \dots m\}$ over the set of resources, and consider the following procedure. We initialize the table with the empty set in all entries. At stage j, for each subset $S \in A$, s.t. $|S| < k$, let $T = A(S \cup \{j\})$. If $|S \cup \{j\}| \leq |T|$ or $T = \emptyset$, we set $A(S \cup \{j\}) = S \cup \{j\}$. After the m^{th} stage we iterate over al entries in the table, and choose the subset with highest social welfare. The procedure runs in $O(m \cdot (\frac{mn}{\epsilon})^n)$ steps, which is polynomial in m and $1/\epsilon$ as required.

Let S^* denote the optimal solution, $S_j^* = \{i \leq j | i \in S\}$. By induction on the stage of the algorithm, we can show that at stage ℓ there is a subset S_ℓ s.t. $S_\ell \in A(S_\ell^*)$, $|S| \leq |S_\ell^*|$ and for every agent i we have that $v_i(S_\ell^*) - v_i(S_\ell) \leq \ell \cdot \frac{\epsilon b}{m \cdot n}$. For $\ell = 1$ the claim is trivial. For a $\ell \leq m$, if $\ell \notin S_\ell^*$, the claim trivially holds from the inductive hypothesis. Otherwise, there is a subset $S_{\ell-1}$, s.t. $|v_i(S_\ell^*) - v_i(S_{\ell-1} \cup \{\ell\})| = |v_i(S_{\ell-1}^* \cup \{\ell\}) - v_i(S_{\ell-1} \cup \{\ell\})| \leq (\ell - 1) \cdot \frac{\epsilon b}{m \cdot n}$ for every i, and $|S_{\ell-1}| \leq |S_{\ell-1}^*|$. If another subset $S' \neq S \cup \{\ell\}$ is stored in $A(S_\ell^*)$ then $|v_i(S \cup \{\ell\}) - v_i(S')| \leq \frac{\epsilon b}{m \cdot n}$, $|S'| \leq |S \cup \{\ell\}|$, and the claim holds. □

MIR Mechanisms

In a similar fashion to the lower bound shown in the previous section, we can show that there is no hope in MIR mechanisms for this class of valuations either.

Theorem 3.3.3. *No computationally-efficient MIR mechanism can approximate CPP with* 2 *capped additive valuations within* $m^{-(\frac{1}{2}-\epsilon)}$ *(for any constant* $\epsilon > 0$*) unless* $NP \subset P/poly$.

Proof. In the previous chapter we showed that even for a single agent with an additive valuation function, an algorithm for CPP which achieves an at least $m^{1/2-\epsilon}$ has a range of size $\Omega(e^{m^\epsilon})$. Since additive functions are a special case of capped additive, the proof holds in this case as well. Again, we can use the Sauer-Shelah lemma to see that the range has a VC dimension at least m^α for some constant $\alpha > 0$.

We now rely on the structure of the reduction of the NP-hardness proof. The number of items which are valued by agent a_1 at 0 and agent a_2 at C_2/k doesn't affect the proof (as long as it is larger than ℓ and at least k), so we just add $m - m^\alpha$ more of these. The particular value of k also doesn't matter, as long as it's at least ℓ, so losing control of how k relates to m^α isn't an issue. Thus, using the same reduction after this modification, we see that the MIR mechanism can be used to solve subset sum instances of size m^α, and is therefore does not run in polynomial time unless $NP \subset P/poly$. □

3.4 Subaddtive Valuations

From a purely computational perspective, we know that for submodular agents there is a $e/(e-1)$-approximation algorithm which is achievable even in the value-queries model. In this section we relax the submodularity assumption. We show both communication and computational complexity lower bounds on the subclass of subadditive valuations known as *fractionally subadditive*. We also show a MIR (and thus truthful) mechanism for subadditive valuations that is a $\sqrt{m}$ approximation. This mechanism makes for a tight upper bound for many of our results from this chapter, and importantly of the main result from the previous chapter.

A Communication Complexity Lower Bound

We now prove that if the valuation functions are subadditive then no constant approximation ratio is possible. Actually, we prove that this holds for a more restricted family of valuation functions called *fractionally-subadditive* [25] (defined in [44] and called XOS there). Informally, a valuation function is fractionally-subadditive if it is the pointwise maximum over a set of additive valuations.

Definition 6. *A valuation function v is said to be fractionally-subadditive if there is a set of additive valuations $\{v_1, ..., v_\ell\}$ such that for every $S \subseteq M$ $v(S) = \max_{r\in[\ell]}\ v_r(S)$.*

Fractionally-subadditive valuations are known to be strictly contained in the class of subadditive valuations and to strictly contain all submodular valuations [44, 37].

Theorem 3.4.1. *Obtaining an approximation ratio of $m^{\frac{1}{4}-\epsilon}$ for fractionally-subadditive valuation functions requires exponential communication (for every $\epsilon > 0$).*

Proof. Fix a small $\epsilon > 0$. We prove the theorem for the case $n = k = \sqrt{m}$. The proof is by reduction from SET DISJOINTNESS (see proof of Lemma 2.4.3 in previous chapter for details). We start by constructing a family of sets $F = \{S_1, ..., S_t\}$ that has the following useful property: Each $S_i \subseteq [m]$, each S_i is of size $m^{\frac{1+\epsilon}{2}}$, and for every two $i \neq j \in [t]$ it holds that $|S_i \cap S_j| \leq 2m^\epsilon$. How big can such an F be? Via the probabilistic method, we show that it can be of size t that is exponential in m.

Let Q, R be two randomly chosen sets of size $m^{\frac{1+\epsilon}{2}}$. For every resource $j \in [m]$ we define a variable X_j that is a assigned a value of 1 if $j \in Q \cap R$ and of 0 otherwise. Observe, that the probability that $X_j = 1$ is $m^{1-\epsilon}$. By using the Chernoff bounds (see Claim 2.4.2) we can show that:

$$Pr[|Q \cap R| > 2m^\epsilon] = Pr[\Sigma_j X_j > 2m^\epsilon] < e^{\frac{-4m^\epsilon}{3}}$$

Since this must hold for any $i \neq j \in [t]$ we get that as long as $t^2 \leq e^{\frac{4m^\epsilon}{3}}$ there is a family F of such size. So, we can set $t = e^{\frac{2m^\epsilon}{3}}$.

Now, we show the reduction from the SET DISJOINTNESS problem. Let $1, 2, ..., \sqrt{m}$ be the parties, and set $t = e^{\frac{m^{2\epsilon}}{3}}$. Let A_i be the subset of $[t]$ held by party i. We identify each element $r \in [t]$ with a set S_r in the family F of subsets of $[m]$ described above. Each party i is now instructed to construct a valuation function v_i in the following manner: Let a_S denote the additive valuation that assigns a value of 1 to every resource in S and a value of 0 to every resource $j \notin S$. Let $v_i = \max\{a_{S_r} | r \in A_i\}$.

Observe that if $\bigcap_i A_i \neq \emptyset$ then there is a set S_r that has a corresponding additive valuation in all of the v_i's. Hence, assigning a subset of S_r of size $\sqrt{m}$ to the users (simulated by the Set Disjointness parties) results in a social welfare value of m. What happens if for every two $i \neq j \in [t]$ $S_i \cap S_j = \emptyset$? We shall now show that in this case the optimal social welfare is $O(m^{\frac{3}{4}+\epsilon})$. This would mean that an approximation of $O(m^{\frac{1}{4}-\epsilon})$ to the CPP problem with fractionally-subadditive valuations enables the distinction between the two extreme cases in the Set Disjointness problem. Therefore, we will then be able to conclude that $\Omega(\frac{t}{n})$ bits are required to do so (a number exponential in both n and m).

So, we are left with showing that if for every two $i \neq j \in [t]$ $S_i \cap S_j = \emptyset$ then the optimal social welfare is $O(m^{\frac{3}{4}+\epsilon})$. Assume, for point of contradiction, that there is some set T of size $\sqrt{m}$ such that the social welfare derived from T, $SW(T)$, is greater than $2m^{\frac{3}{4}+\epsilon}$. Let a_i be an additive valuation function of i for which v_i is maximized (for T). Observe that $SW(T) = \Sigma_{i\in[n]}\ a_i(T)$. Assume, w.l.o.g., that $T = \{1, ..., \sqrt{m}\}$. For every resource $j \in T$, let x_j be the number of the a_i's that assign a value of 1 to j. Observe that $SW(T) = \Sigma_{i\in[n]}\ a_i(T) = \Sigma_{j\in T}\ x_j$. Also observe that $\Sigma_{j\in T}\ x_j(x_j - 1) = \Sigma_{i\neq i'}|S_i \cap S_{i'} \cap T|$. We now have that:

$$2m^{1+\epsilon} = 2n^2m^\epsilon \geq \Sigma_{i\neq i'}|S_i \cap S_{i'} \cap T| = \Sigma_{j\in T}\ x_j(x_j - 1) \tag{3.1}$$

this is due to the fact that the cardinality of the intersection of every two S_is cannot exceed $2m^\epsilon$. Using elementary calculus, it is easy to show that $\Sigma_{j\in T}\ x_j^2 \geq m^{\frac{1}{4}}\Sigma_{j\in T}\ x_j$, due to the fact that the worst case ratio is achieved when all x_j's are equal. Combining the last two equations gives us that:

$$SW(T) = \Sigma_{j\in T}x_j \leq 2m^{\frac{3}{4}+\epsilon}. \tag{3.2}$$

A contradiction. □

Computational Complexity Lower Bound

In the previous section we discussed fractionally-subadditive functions in the communication complexity model, and assumed that a fractionally-suabadditive valuation can be represented as the maximum of exponentially many additive valuations. In this section we are interested in understanding the *computational complexity* limitations of valuations with such structure that have succinct representation. We will consider the subclass of *succinctly described* fractionally-subadditive valuations: fractionally-subadditive valuations that can be represented using *polynomially* (in m) many additive valuations. That is, we will consider valuations $v : 2^{[m]} \to \mathbb{R}_+$ that can be represented as the maximum of $O(poly(m))$ additive valuations.

We first show that there is a polynomial time algorithm for succinctly described fractionally-subadditive agents when the number of agents is *constant*.[1]

Theorem 3.4.1. *CPP with a constant number of succinctly described fractionally-subadditive valuations can be solved in polynomial time.*

Proof. Each fractionally-subadditive valuation is the maximum over polynomially many additive valuations. If one of these additive valuations is chosen for each agent, the resulting public project problem can be trivially solved in polynomial time. This solution gives a lower bound on the maximum social welfare. If the additive valuations chosen happen to be the ones that exhibit the maximum in an optimal allocation, the solution found will also be optimal. Thus, by enumerating over all possible choices, an optimal allocation can be found. If there are c agents with at most ℓ additive valuations each, there are $O(\ell^c) \subseteq poly(\ell)$ choices to enumerate over. Thus, the solution to the auction can be found in polynomial time. □

We now give a reduction from LABEL COVER$_{max}$ to CPP with n fractionally-subadditive valuations which preserves an approximation gap. First, we define LABEL COVER$_{max}$ and discuss the complexity of its approximation. A LABEL COVER$_{max}$ instance consists of a regular bipartite graph $G = (V_1, V_2, E)$, a set of n labels $N = \{1, \dots, n\}$ and for each edge $e \in E$ a partial function $\Pi_e : N \to N$. We say that the edge $e = \{x, y\}$ for $x \in V_1, y \in V_2$ is satisfied if x is labeled with l_1 and y with l_2 such that $\Pi_e(l_1) = l_2$. The goal of LABEL COVER$_{max}$ is to find an assignment of labels to the nodes in V_1 and V_2 such that each node has exactly one label and as many edges as possible are satisfied. It was shown in [5] that LABEL COVER$_{max}$ is quasi-NP-hard to approximate.

Theorem 3.4.2 ([5])**.** *For any sufficiently small constant $\gamma > 0$, it is quasi-NP-hard to distinguish between the following two cases in* LABEL COVER$_{max}$*: (1) YES case: all edges are covered, and (2) NO case: at most a $2^{-\log^{1-\gamma} n}$ fraction of the edges are covered, where n is the size of the* LABEL COVER$_{max}$ *instance.*

We make use of Theorem 3.4.2 to show a similar hardness result for CPP with fractionally-subadditive valuations.

Theorem 3.4.3. *Obtaining an approximation ratio of $2^{\frac{\log^{1-\gamma} b}{6}}$ for CPP with fractionally-subadditive valuations where b is the size of the CPP instance is quasi-NP-hard.*

Proof. We prove this using a gap-preserving reduction from LABEL COVER$_{max}$: We are given an instance of LABEL COVER$_{max}$ consisting of a graph $G = (V_1, V_2, E)$, a set of labels N and a set of partial functions Π_e for each $e \in E$. We create a CPP instance with $|V_1|$

[1]Note that the fact that the valuations are succinctly described implies that this class *does not* include all submodular functions. In particular, while it is NP-hard to approximate even a single submodular valuation within a factor better than $e/(e-1)$, in the case of succinctly described fractionally-subadditive valuations we can obtain an optimal solution for a constant number of agents in polynomial time.

agents, one corresponding to each node in V_1. The resource set is $V_2 \times N$. We now define the fractionally-subadditive valuation v_i of each agent i. For every label $l \in N$, we define the additive valuation function $a_{i,l}$.

$$a_{i,l}(\{(j,l')\}) = \begin{cases} 1, & \{i,j\} \in E \text{ and } \Pi_{\{i,j\}}(l) = l' \\ 0, & \text{otherwise} \end{cases}.$$

So $a_{i,l}(S)$ represents how many edges incident with i are covered if we choose label l for vertex $i \in V_1$ and the best label from the set $\{l' : (j,l') \in S\}$ for vertex $j \in V_2$.

The fractionally-subadditive valuation of agent i is defined by

$$v_i(S) = \max_{l \in N}\{a_{i,l}(S)\}.$$

So agent i gets the value for the best possible choice of a single label for vertex i given the label choices for V_2 implied by S. We set the size of the set of resources to be chosen in our CPP instance to be $|V_2|$.

If the Label Cover$_{max}$ instance is a YES case, we can find a set of resources with social welfare $|E|$. Simply take any labeling that covers every edge and for every $j \in V_2$, choose the resource (j,l'), where j is labeled by l' in the labeling. Call this set S. Clearly, $v_i(S)$ equals the degree of node i, as if we choose l such that i is labeled by l, $\Pi_{\{i,j\}}(l) = l'$ for each $(j,l') \in S$. So the social welfare given these resources is $|E|$.

We now show that if the Label Cover$_{max}$ instance is a NO case, then the maximum social welfare is bounded by $2^{-\frac{\log^{1-\gamma} n}{6}}|E|$ for sufficiently large n. Note that if n' is the size of the Label Cover$_{max}$ instance, our construction guarantees $n \leq (n')^2$. So our bound is at least

$$2^{-\frac{\log^{1-\gamma}[(n')^2]}{6}}|E| \geq 4 \cdot 2^{-\frac{\log^{1-\gamma} n'}{3}}|E|$$

for sufficiently large n'. In order to simplify our expressions in the rest of the proof, let $\alpha = 2^{-\frac{\log^{1-\gamma} n}{6}}$. Using the above bound, we see

$$\alpha \geq 4 \cdot 2^{-\frac{\log^{1-\gamma} n'}{3}}. \tag{3.3}$$

Suppose by way of contradiction that we reduced from a NO case, but the maximum social welfare is at least $\alpha|E|$.

Let S be a set of resources in the CPP instance with a social welfare of at least $\alpha|E|$. Recall also that each agent i's fractionally-subadditive valuation v_i is defined as the pointwise maximum over a set of additive valuations. Let $a_{i,l}$ be the additive valuation in the set of valuations making up v_i for which $a_{i,l}(S)$ is maximized (and so $v_i(S) = a_{i,l}(S)$). If we fix a choice of j, $a_{i,l}$ assigns a value of 1 to at most one of the resources (j,l') for $l' \in N$. Moreover, $a_{i,l}$ can only assign value to a resource (j,l') if $\{i,j\} \in E$. We say that an edge between vertex $i \in V_1$ and vertex $j \in V_2$ is satisfied by the set S if $(j, \Pi_{\{i,j\}}(l)) \in S$. Observe that the total social welfare value of S equals the number of edges satisfied by S.

Let d be the number of incoming edges of a vertex in V_2. Since G is a regular bipartite graph, $d = \frac{|E|}{|V_2|}$. Let V_2' denote all vertices $v \in V_2$ in which the number of edges incident on v satisfied by S is at least $\frac{\alpha}{2}d$. A counting argument shows that $|V_2'| \geq \frac{\alpha}{2}|V_2|$. If $|V_2'|$ were less than $\frac{\alpha}{2}|V_2|$, the number of satisfied edges incident upon vertices in V_2' is at most $|V_2'|d < \frac{\alpha}{2}|E|$, and the number of satisfied edges incident upon vertices outside of V_2 would be less than $|V_2|\frac{\alpha}{2}d = \frac{\alpha}{2}|E|$. So summing these, we would see that the number of satisfied edges is less than $\alpha|E|$, a contradiction. So $|V_2'| \geq \frac{\alpha}{2}|V_2|$.

If S contains ℓ resources of the form (j, l) for a fixed j and ℓ different values $l \in N$, we say that j is labeled ℓ times by S. Since there are $|S| = |V_2|$ resources, at most $\frac{\alpha}{4}|V_2|$ of the nodes $j \in V_2$ are labeled more than $\frac{4}{\alpha}$ times by S. So letting V_2'' be the subset of V_2' which is labeled at most $\frac{4}{\alpha}$ times by S, $|V_2''| \geq \frac{\alpha}{4}|V_2|$.

Since S labels each $j \in V_2''$ at most $\frac{4}{\alpha}$ times, and S satisfies at least $\frac{\alpha}{2}d$ edges incident upon each vertex in V_2'', we can find a single $s_j \in S$ that satisfies at least $\frac{\alpha/2}{4/\alpha}d = \frac{\alpha^2}{8}d$ of the edges incident upon j. So if we label each $j \in V_2''$ according to S_j and label each $i \in V_1$ by the l such that $v_i(S) = a_{i,l}(S)$, we have a labeling that satisfies at least $|V_2''|\frac{\alpha^2}{8}d = \frac{\alpha^3}{32}|E|$ edges, regardless of how the vertices in $V_2 - V_2''$ are labeled. This contradicts that we had a NO case, as we can see by (3.3) that $\frac{\alpha^3}{32}|E| > 2^{-\log^{1-\gamma} n'}|E|$.

Thus, we see that the maximum social welfare of our CPP is at least $|E|$ if we reduced from a YES case and at most $\alpha|E|$ if we reduced from the NO case. Therefore it is quasi-NP-hard to achieve an approximation ratio of $\alpha = 2^{-\frac{\log^{1-\gamma} n}{6}}$. $\square$

A Truthful $\sqrt{m}$ Approximation Algorithm

We show that the impossibility result from the previous chapter is tight by presenting a simple truthful algorithm which obtains a $\min\{k, \sqrt{m}\}$ approximation ratio (for any value of k and n) and requires at most $n \cdot m$ value queries. As shown by the characterization lemma in the previous chapter, any truthful algorithm for CPP with submodular agents must be an affine maximizer (which is MIR). Indeed, the algorithm presented in this subsection is a simple MIR mechanism.

A Deterministic Mechanism for Subdditive valuations

1. Arbitrarily partition $[m]$ into $r = \max\{\frac{m}{k}, \sqrt{m}\}$ disjoint subsets of equal size $S_1, ..., S_r$.
2. Ask each agent to specify her value for each of the different subsets S_t.

Output: Choose the subset S_t that maximizes the social welfare $\Sigma_i v_i(S_t)$

Since this algorithm is MIR we know that it can be made truthful via VCG payments [60, 16, 30]. Observe that the algorithm indeed requires at most m value queries to be addressed to each of the n users. Therefore, all that is left to show is that the algorithm provides the required approximation-ratio. We show that this is true even if users' valuation functions are subadditive.

Proposition 3.4.2. *If $v_1, ..., v_n$ are subadditive then the algorithm provides an approximation ratio of* $\min\{k, \sqrt{m}\}$.

Proof. Let S^* be a set of size k that maximizes the social welfare. First consider the case where $k \leq \sqrt{m}$. Then, by (iterative use of) subadditivity, for every $i \in [n]$, $v_i(S^*) \leq \Sigma_{j \in S^*} v_i(\{j\})$. Hence, $\Sigma_{i\in[n]} v_i(S^*) \leq \Sigma_{i\in[n]}\Sigma_{j\in S^*} v_i(\{j\}) = \Sigma_{j\in S^*}\Sigma_{i\in[n]} v_i(j)$. This implies that there is an element $j \in [m]$ such that the social welfare derived from j is at least $\frac{1}{|S^*|} = \frac{1}{k}$ of the optimal social welfare. This item j appears in one of the S_t's, and so, because the valuations are non-decreasing, the social welfare derived from that S_t is also at least $\frac{1}{k}$ of the optimal social welfare. Since the algorithm optimizes over all the S_t's it is bound to achieve the desired approximation ratio.

Consider the case where $k > \sqrt{m}$. Because the valuations are non-decreasing, $\Sigma_i v_i(S^*) \leq \Sigma_i v_i([m])$. Let $S_1, ..., S_{\sqrt{m}}$ be some arbitrary partition of $[m]$ into $\sqrt{m}$ disjoint subsets of size $\sqrt{m}$. Exploiting subadditivity in a way similar to that shown above implies that for one of these sets the social welfare is at least a $\frac{1}{\sqrt{m}}$ fraction of the social welfare for the entire set $[m]$. This concludes the proof of the proposition. $\square$

3.5 General Valuations

In this section we study CPP with general valuations (but still normalized and non-decreasing). We prove strong inapproximability results with general valuations in both the computational and the communication-complexity models. In the communication-complexity model our lower bound is tight (a trivial matching upper bound exists).

Theorem 3.5.1. *Obtaining an approximation of $O(n^{\frac{1}{2}-\epsilon})$ to the social welfare in CPP with general valuations, for any $\epsilon > 0$, is impossible unless $P = NP$. Obtaining an approximation of $O(n^{1-\epsilon})$ to the social welfare is impossible unless $P = ZPP$.*

Proof. We reduce from the Maximal Welfare Tree (MWT) problem studied in the context of distributed algorithmic mechanism design [27, 28, 38]. Our reduction preserves the hardness results for this problem as shown in [38]. In the MWT problem we are given a graph $G = (N, L)$ with a set of nodes N and links L. A unique destination node d is given and each node $a \in N \setminus \{d\}$ has a valuation function $v_a : P_a \to \mathcal{R}_{\geq 0}$, where P_a is used to denote the set of all simple paths from a to the destination d. The objective in MWT is to form a tree rooted in d which maximizes the social welfare, i.e., choose the tree T^* such that $T^* \in \text{argmax}_{T\in\mathcal{T}_L^d} \sum_{a\in N\setminus\{d\}} v_a(T)$, where $\mathcal{T}_L^d$ is the set of all possible trees in L rooted in d. We consider the special case of MWT in which for all $a \in N \setminus \{d\}$ we have $v_a : P_a \to \{0, 1\}$. It is known that for any $\epsilon > 0$ approximating MWT, even for this special case, within a factor of $O(n^{\frac{1}{2}-\epsilon})$ is impossible unless $P = NP$ and approximating within a factor of $O(n^{1-\epsilon})$ is impossible unless $P = ZPP$ [38].

The reduction from MWT is as follows: Given an instance of MWT such that the range of all valuation functions is $\{0,1\}$, for each link $\ell \in L$ we associate a resource ℓ' in CPP and each node $a \in N \setminus \{d\}$ in MWT will correspond to an agent a' in CPP. It remains to define the valuation function of a'. Note that since our interest is in showing a lower bound, we can adversarially set the number of chosen items to be $k = |N| - 1$. Now, let P_a^+ be the set of paths for which $v_a = 1$. For all $E \subseteq L$, the valuation function for the corresponding agent a' in CPP is defined by:

$$v_{a'}(E) = \begin{cases} 1 & \exists P \in P_a^+ : P \subseteq E \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Observe that choosing a tree T in MWT with social welfare value $SW(T) = c$ corresponds to choosing a set of resources that induces the same social welfare value in CPP. Conversely, choosing a set of resources T' in CPP s.t. $SW(T') = c'$ necessarily means that we can trim T' to a set of edges T which forms a routing tree with d as its source, and that we have exactly c' nodes which have routes to d in T, and hence $SW(T) = c'$ in MWT. □

Theorem 3.5.2. *Obtaining an approximation ratio of* $(1-\epsilon)n$ *for general valuations requires exponential communication in* m *(for any* $\epsilon > 0$ *and for any* $n << 2^m$*).*

Proof. For CPP with general valuation functions, n agents, m items and a parameter $1 \leq k \leq m$ we show a lower bound of $\Omega(\binom{m}{k} \cdot n^{-1})$ again by reducing from the Set Disjointness problem.

We construct an instance of CPP with n agents in which no restrictions (except for being normalized and non-decreasing) apply to the agents' valuation functions. Let $S_1, \ldots, S_t$ be the (ordered) sets in the range of all possible allocations of size k. For each party i in Set Disjointness with the set $A_i \subseteq \{1, \ldots, t\}$, we associate an agent i in CPP with the following valuation function:

$$v_i(S_r) = \begin{cases} 1 & r \in A_i \\ 0 & \text{otherwise} \end{cases}$$

Observe that these valuation functions are indeed normalized and non-decreasing. Let S_l be the set which maximizes the social welfare, i.e., $\ell \in \text{argmax}_{\ell \in [d]} \ |\{A_i| \ l \in A_i\}|$. To approximate the social welfare within a factor of $(1-\epsilon)n$ for any $\epsilon > 0$, one must allocate some set S for which there are at least two agents i and j such that $v_i(S) = v_j(S) = 1$. Due to the above construction of the agents' valuation functions this necessarily implies deciding between the two extreme cases of the Set Disjointness problem. Thus, for $d = \binom{m}{k}$ we get a lower bound of $\Theta(\binom{m}{k} \cdot n^{-1})$ for CPP with general valuation functions. □

In the communication model a trivial matching upper bound of n exists: Query each agent i for her most valued set S_i of size k, and assign the agents a set $T \in \text{argmax}_{i \in [n]} v_i(S_i)$. It is easy to see that this indeed guarantees an n-approximation.

3.6 Discussion

The results in this chapter highlight the richness and depth of the combinatorial public projects model. As we saw, the problem becomes NP-hard for very strict valuations classes (unit demand with n agents, and capped additive with two agents), and there is a broad class of valuations where the VCG mechanism cannot be applied. We know that MIR mechanisms are hopeless in these restricted classes, though interestingly, these are not the only truthful mechanisms we can expect. The natural question is whether there are non-MIR mechanisms for classes such as capped additive that can obtain reasonable approximation ratios. For cases of subadditive functions, we saw impossibility results that are strictly due to computational limitations. This implies that the barrier for implementation in these classes is computation and not incentive compatibility.

Part II

Possibilities of Algorithmic Mechanism Design

Chapter 4

Budget Feasible Mechanisms I: How To Win Friends and Influence People, Truthfully

4.1 Introduction

In the previous part we focused on the tension between incentive compatibility and computational constraints. The bottleneck for executing incentive compatible mechanisms however, is not always computation. In many emerging online markets the main barrier to implementation is the potential overpayments associated with incentive compatibility. Consider the following example.

Influence Maximization in Social Network. In social network marketing the goal is to incentivize a small set of individuals in a social network to recommend a product, in a way that maximizes the word-of-mouth effect in the network. The market designer is naturally limited by the amount of rewards it can offer, and each individual has a different cost for making a recommendation to her friends in the network. Since individuals may lie about their cost, the market designer strives to design an incentive compatible mechanism that will maximize the word-of-mouth effect in the network. The main problem, however, is that the incentive compatible payments must be under the budget.

Budget Feasible Mechanisms

The social marketing problem above is an example of markets that procure information and services from agents and aim to optimize complex objectives under a budget. While incentive compatibility is the golden standard for implementation, there is a clash between budget and incentives: Classical mechanism design theory rules out incentive compatibility under a budget as it is well known that implementing optimal solutions – even for very simple objectives – can result in huge overpayments [7].

In the following chapters we will develop a theoretical framework that enables designing incentive compatible mechanisms under a budget. Although the main difficulties are not computational, we will use a computational approach to deal with the impossibilities of designing incentive compatible mechanisms under a budget by resorting to *approximations*.

In the settings we discuss there is a single buyer and many agents. Each agent has a single item or service they are selling, for which they associate some private cost. The buyer has a utility function over subsets of items and a budget and wishes to buy items that maximize her utility function under the budget. Before discussing the model formally we emphasize two main points.

- **Approximation is necessary.** Consider the case where the buyer's goal is to buy as many items possible under a budget constraint. The standard framework in mechanism design calls for implementing the optimal solution of the full-information problem and paying agents in a manner that supports incentive compatibility. In case all items have small costs $\epsilon > 0$ and the budget is B, the optimal solution will allocate to all the agents and, according to Myerson's characterization of threshold payments, will pay each agent $B - \epsilon n$. The solution returned by the mechanism is indeed optimal, and incentive compatible, but the payments exceed the budget by a factor of n. This is the VCG mechanism. The message here is twofold. First, the VCG mechanism is not a feasible alternative in this case, not due to computational constraints but due to its overpayments. Second, this example shows that, in general, implementing the optimal solution is not feasible.

- **Superadditivity implies bad approximations.** Consider a slight variation of the above problem, in which all items have small costs, and identical values *as long as a particular item i is in the solution*, and otherwise all have value 0 (for example, think of i as a corkscrew and the rest of the items as bottles of wine). How well can a budget feasible mechanism do here? If the mechanism has a bounded approximation ratio it must always guarantee to include i in its solution. This however implies that as long as i declares a cost that is less than the mechanism's budget, the mechanism includes her in the solution. A truthful mechanism must therefore surrender its entire budget to i. This of course results in an unbounded approximation ratio.

The above examples help in phrasing the question that will guide the following chapters. The first example above shows that the optimal solution is infeasible, due to overpayments. To circumvent this, we must therefore resort to *approximations*. The important point is that approximation is used due to the clash between budget and incentive compatible payments, and not due to computational limitations. The second example shows that there are (even very simple) classes of buyer objectives, for which no mechanism under a budget can yield good approximation. We use the term *budget feasible* to describe a mechanism whose sum of payments are below a given budget. The question, then, is:

Which procurement settings have budget feasible mechanisms with desirable guarantees?

In single parameter domains, where each agent's private information is a single number, designing truthful mechanisms often reduces to designing monotone allocation rules, since payments can be computed via binary search [41]. This no longer holds when the payments are restricted by a budget: Designing a budget feasible allocation rule requires understanding its payments, which in-turn depend on the allocation rule itself. Not surprisingly, it seems that budget feasible mechanisms are very tricky to find.

The main result in this chapter shows that budget feasible mechanisms with desirable guarantees are obtainable in *submodular* procurement markets.

Theorem. *For any procurement where the objective is an increasing submodular utility function there exists a randomized mechanism which is budget feasible and universally truthful, and in expectation provides a constant-factor approximation of the optimal solution.*

From a computational perspective, this result is somewhat tight as for a slightly more general class of utility functions there is an information theoretic lower bounds as we discuss in the next chapter. In the following chapter we also discuss other classes of utility functions, and present mechanisms that build on the main ideas presented in this chapter.

Related Work

Budgets came under scrutiny in auction theory after observing behavior of bidders in online automated auctions [2, 18, 14, 29, 12], as well as in spectrum auctions where bidding is performed by groups of strategic experts [14]. While these works highlight the significance and challenges that budgets introduce to mechanism design, they relate to an entirely different concept than the one we study here since they examine a budget on the bidders and not the mechanism's payments.

In recent years a theory of *frugality* has been developed with the goal of providing mechanisms for procurement auctions that admit minimal payments [15, 4, 32, 23, 58] . Budget feasibility and frugality are complementary concepts. Frugality is about buying a feasible solution at minimum cost — there are no preferences among the solutions, and the goal is to minimize payment. In our setting we have no preferences among payments — as long as they are below the budget — but we do care about the value of the solutions. The two approaches are complementary also in another important sense: in our last section we show that for all the problems studied in the frugality literature there are no budget feasible mechanisms. We discuss this point in further detail in Section 4.5.

Organization of the Chapter

We begin with a formal presentation of the model in Section 4.2. As a warm up, we discuss the class of symmetric submodular functions in Section 4.3; this special case simplifies the problem enormously and facilitates the introduction of ideas and intuition for the general submodular case. Our main result for submodular functions is developed in Section 4.4. We conclude with a few simple results that develop our understanding of the budget feasibility model in Section 4.5.

4.2 The Model

In a *budget-limited reverse auction* we have a set of items $\mathcal{N} = \{1, \ldots, n\}$, and a single buyer. Each item $i \in [n]$ has a cost $c_i \in \mathbb{R}_+$, while the buyer has a budget $B \in \mathbb{R}_+$ and a utility function $f : 2^{[n]} \to \mathbb{R}_+$. In the *full information* case costs are common knowledge, and the objective is to maximize the utility function under the budget, i.e. find the subset $S \in \{T | \sum_{i \in T} c_i \leq B\}$ for which $f(S)$ is maximal.

We focus on the *strategic case*, in which each item is held by a unique agent and costs are *private*. We use a_i to denote the agent that is associated with the item $i \in \mathcal{N}$. In cases where it is clear from the context we will simply use i to denote agent a_i. The budget and utility function of the buyer are common knowledge. A solution is a subset of agents and a payment vector, and the objective is to maximize the utility function while the *payments* (not costs) are within the budget.

More formally, a mechanism $\mathcal{M} = (\mathcal{A}, p)$ consists of an allocation function $\mathcal{A} : \mathbb{R}^n_+ \to 2^{[n]}$ and a payment function $p : \mathbb{R}^n_+ \to \mathbb{R}^n_+$. The allocation function $\mathcal{A}$ maps a set of n bids to a subset $S = \mathcal{A}(c_1, \ldots, c_n) \subseteq [n]$. The payment function p returns a vector $p_1, \ldots, p_n$ of payments to the agents. We shall often omit the arguments $c_1, \ldots, c_n$ when writing $\mathcal{A}$ and p. We will denote by $s_1, \ldots, s_n$ the characteristic vector of S, that is, $s_i = 1$ iff $i \in S$. As usual, we seek normalized ($s_i = 0$ implies $p_i = 0$), individually rational ($p_i \geq s_i \cdot c_i$) mechanisms with no positive transfers ($p_i \geq 0$). As it is common in algorithmic mechanism design, our goal is manifold. We seek mechanisms that are:

1. **Truthful.** That is, reporting the true costs is a dominant strategy for sellers. Formally, a mechanism $\mathcal{M} = (\mathcal{A}, p)$ is *truthful* (*incentive compatible*) if for every $i \in \mathcal{N}$ with cost c_i and bid c'_i, and every set of bids by $\mathcal{N} \setminus \{a_i\}$ we have $p_i - s_i \cdot c_i \geq p'_i - s'_i \cdot c_i$, where (s_i, p_i) and (s'_i, p'_i) are the allocations and payments when the bidding is c_i and c'_i, respectively. A mechanism that is a randomization over truthful mechanisms is *universally truthful*.

2. **Computationally Efficient.** The functions $\mathcal{A}$ and p can be computed in polynomial time. In cases where the utility function requires exponential data to be represented (as in the general submodular case for example), we take the common "black-box" approach and assume the buyer has access to an oracle which allows evaluating any subset $S \subseteq [n]$, with polynomially many queries. Such queries, as discussed in the previous chapters, are known as value queries. This is a weaker model than ones allowing demand or general queries (see [11] for a definition) and since our main interest here is algorithmic, this strengthens our results. In the following chapter we will discuss mechanisms that use demand oracles.

3. **Good.** We want the allocated subset to yield the highest possible value for the buyer. For $\alpha \geq 1$ we say that a mechanism is α-approximate if the mechanism allocates to a set S such that $OPT \leq \alpha f(S)$, where OPT denotes the value of full information optimal solution that can be achieved without computational constraints. As usual, when dealing with randomization we seek mechanisms that yield constant factor approximations in expectation.

4. **Budget Feasible.** Importantly, we require that a mechanism's allocation rule and payments do not exceed the budget: $\sum_i p_i s_i \leq B$. We call such mechanisms *budget feasible.*

Observe that this is a *single parameter* mechanism design problem, in that the private value of each bidder can be represented as a single real number. We shall repeatedly rely on Myerson's well-known characterization for truthful mechanisms in these domains.[1]

Theorem 4.2.1 ([42]). *In single parameter domains a normalized mechanism $\mathcal{M} = (\mathcal{A}, p)$ is truthful iff:*

(i) ***$\mathcal{A}$ is monotone:*** *$\forall i \in [n]$, if $c'_i \leq c_i$ then $i \in \mathcal{A}(c_i, c_{-i})$ implies $i \in \mathcal{A}(c'_i, c_{-i})$ for every c_{-i};*

(ii) ***winners are paid threshold payments:*** *payment to each winning bidder is $\inf\{c_i : i \notin \mathcal{A}(c_i, c_{-i})\}$.*

4.3 Symmetric Submodular Functions

We begin by introducing a simple subclass of submodular functions which is devoid of many of the intricacies of the general case. It will serve as an exposition of the basic ideas, and help explain the main difficulties.

We say a set function is *symmetric* if it only depends on the cardinality of the set, rather than the identity of the items it holds. Symmetric submodular functions (also called *downward sloping*), were used by Vickrey in his seminal work on multi-unit auctions [60]. They have a very simple structure:

Definition 7. *A function $f : 2^{[n]} \to \mathbb{R}_+$ is symmetric submodular if there exist $r_1 \geq \ldots \geq r_n \geq 0$, such that $f(S) = \sum_{i=1}^{|S|} r_i$.*

Consider the following allocation rule $\mathcal{A}$: Sort the n bids so that $c_1 \leq c_2 \leq \ldots \leq c_n$, and consider the largest k such that $c_k \leq B/k$. That is, k is the place where the curve of the

[1]Note that although there is a budget constraint on the payments, Myerson's characterization applies to our setting as well. Due to the characterization, we know that the allocation function determines the payment function. The budget constraint can therefore be viewed as a property of the allocation function alone.

increasing costs intersects the hyperbola B/k. The set allocated here is $S = \{1, 2, \dots, k\}$. This is a monotone allocation rule: an agent cannot be excluded when decreasing her bid. While the natural candidate for the threshold payment for this allocation rule seems to be B/k for each agent in $[k]$, the example show it is not enough.

Example: The payment rule $p_i = B/k$ breaks truthfulness. Consider running the mechanism shown in section 4.3 with payments B/k on three agents with real costs $c_1 = 3, c_2 = 5 - \epsilon, c_3 = 5$ and a budget $B = 10$. If the mechanism were truthful, then it should result in agents with real costs 3 and $5 - \epsilon$ being allocated, and paid 5. Note that in this case it would be in the best interest of agent a_3 to report a false cost $c'_3 = 5 - 2\epsilon$ since, in this case, the allocation would go to her and agent a_1, leaving a_3 with a profit of ϵ. Paying the minimum of the fair share and the cost of the agent that is excluded from the solution solves this.

Proposition 4.3.1. *The mechanism with the allocation $\mathcal{A}$ as above with payments of $\theta_i = \min\{B/k, c_{k+1}\}$ to every agent a_i in its allocated set is truthful.*

Proof. The allocation rule is monotone since declaring a lower cost advances an item in the sorting. Let $S = \{1, 2, \dots, k\}$ be the set of agents allocated by the mechanism. To see that θ_i is indeed the threshold payment for all $a_i \in S$, consider first the case where $c_{k+1} < B/k$. Declaring a cost $c'_i > c_{k+1}$ places i in a position after agent a_{k+1}. Since all agents in $(S \setminus \{a_i\}) \cup \{a_{k+1}\}$ have costs less than B/k, as the mechanism reaches agent a_i's bid, there are already (at least) k agents ahead of a_i. Since $c'_i > c_{k+1} > B/(k+1)$ agent a_i will not be allocated. Declaring a cost below c_{k+1} places a_i within the first k items, all with costs less than B/k, and thus i will be allocated.

In case $B/k \leq c_{k+1}$, declaring cost $c'_i > B/k$ places at least $k-1$ items ahead of a_i, since all items in the winning set have cost less than B/k. Therefore, even if a_i will be considered by the mechanism it will not be allocated as it does not meet the mechanism's allocation condition. Declaring a lower cost ensures that a_i is placed within the first k items and it will be allocated. The payment rule therefore respects the threshold property and we conclude that the mechanism is indeed truthful. □

Observe that this allocation rule has the property we seek: summing over the payments that support truthfulness satisfies the budget constraint. Hence this gives us a budget feasible mechanism. Importantly, this is also a good approximation of the optimal solution:

Theorem 4.3.1. *The above mechanism has approximation ratio of two.*

Proof. Observe that the optimal solution is obtained by greedily choosing the lowest-priced items until the budget is exhausted. By the downward sloping property, to prove the result it suffices to show that the mechanism returns at least half of the items in the greedy solution. Assume for purpose of contradiction that the optimal solution has ℓ items, and the mechanism returns less than $\ell/2$ items. It follows that $c_{\lceil \ell/2 \rceil} > 2B/\ell$. Note however, that this is impossible since we assume that $c_{\lceil \ell/2 \rceil} \leq \ldots \leq c_\ell$, and $\sum_{i=\lceil \ell/2 \rceil}^{\ell} c_i \leq B$ which implies that $c_{\lceil \ell/2 \rceil} \leq 2B/\ell$, a contradiction. □

We now show that no better approximation ratio is possible. This illustrates the intricacies of budget feasibility and is rather surprising, given the simplicity of the full-information problem.

Proposition 4.3.2. *For $f(S) = |S|$, no budget feasible mechanism can guarantee an approximation of $2 - \epsilon$, for any $\epsilon > 0$.*

Proof. Suppose we have n items with costs $c_1 = c_2 = \cdots = c_n = B/2 + \epsilon$, for some positive $\epsilon < B/2$. Assume, for purpose of contradiction that $\mathcal{M}$ is a budget feasible mechanism with approximation ratio better than 2. In particular, $\mathcal{M}$ has a finite approximation ratio and must therefore allocate to at least one agent in this case. W.l.o.g., assume $\mathcal{M}$ allocates to agent a_1.

By monotonicity, agent a_1 can reduce her cost to $c'_1 = \epsilon' < B/2 - \epsilon$ and remain allocated. For this cost vector, (c'_1, c_{-1}), Myerson's characterization implies that the threshold payment for agent a_1 should be at least $B/2 + \epsilon$, by individual rationality and budget feasibility, $\mathcal{M}$ cannot allocate to any other agent. Observe however that the optimal full information solution in this case allocates to two agents which contradicts $\mathcal{M}$'s approximation ratio guarantee. □

In summary, the above propositions together with Theorem 4.3.1 show that for symmetric submodular utility functions the proportional share mechanism is the optimal budget feasible mechanism.

Theorem 4.3.2. *For symmetric submodular functions, there exists a truthful budget feasible 2-approximation mechanism. Furthermore, no budget feasible mechanism can do better.*

4.4 General Submodular Functions

We now turn to the general case of submodular functions. Recall that $f : 2^{[n]} \to \mathbb{R}_+$ is submodular if $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T) \quad \forall S \subseteq T$. A function f is *increasing* if $S \subseteq T$ implies $f(S) \leq f(T)$. Throughout this chapter we assume the function is increasing. In the next chapter we will discuss the case of non-increasing submodular functions. In general, submodular functions may require exponential data to be represented. We therefore assume the buyer has access to a *value oracle* which given a query $S \subseteq [n]$ returns $f(S)$. We consider a mechanism to be efficient if it runs in polynomial time and makes a polynomial number of value queries to the oracle. An important concept for our discussion is the marginal contribution of an agent, as defined below.

Definition 8. *The marginal contribution of an agent a_i given a subset S, denoted $f_{i|S}$ is $f(S \cup \{a_i\}) - f(S)$.*

In designing truthful mechanisms for submodular maximization problems, the greedy approach is a natural fit, since it is monotone when agents are sorted according to their increasing marginal contributions relative to cost: In the marginal contribution-per-cost sorting the agent that appears in position $i+1$ is the agent a_j for which $f_{j|S_i}/c_j$ is maximized over all agents $\mathcal{N}$ where $S_i = \{1, 2 \ldots, i\}$, and $S_0 = \emptyset$. To simplify notation, when we consider this sorting and it will be clear from the context we will write f_i instead of $f_{i|S_{i-1}}$. This sorting, in the presence of submodularity, implies:

$$\frac{f_1}{c_1} \geq \frac{f_2}{c_2} \geq \ldots \geq \frac{f_n}{c_n} \tag{4.1}$$

Notice that $f(S_k) = \sum_{i \leq k} f_i$ for all k.

The Proportional Share Allocation Rule

The mechanism from the previous section for the limited symmetric case can be generalized appropriately to work for various classes in the submodular family of functions.

Definition 9. *For a budget B and set of agents $\mathcal{N}$, the generalized proportional share allocation rule, denoted $\mathcal{A}(\mathcal{N}, B)$ sorts agents in $\mathcal{N}$ according to (4.1) with budget B and allocates to agents $i \in \{1, \ldots, k\}$ that respect $c_i \leq B \cdot f_i/f(S_i)$.*

For concreteness consider the case of additive functions: each agent a_i is associated with a fixed value v_i and $f(S) = \sum_{i \in S} v_i$. Here the marginal contribution of each agent is independent of their place in the sorting, and we simply have that $f_i = v_i$ for all agents $a_i \in \mathcal{N}$. In this case $\mathcal{A}$ produces a budget-feasible mechanism. The reason is, it assures us that for each agent a_i, the threshold payments of $\mathcal{A}$, denoted θ_i do not exceed the agent's proportional share. The threshold payments are a generalization of the payments we presented in the symmetric submodular case:

$$\theta_i = \min\left\{\frac{v_i \cdot B}{\sum_{i \in S} v_i}, \frac{v_i \cdot c_{k+1}}{v_{k+1}}\right\}.$$

This allows budget feasibility since $\sum_i \theta_i \leq B$, as well as individually rationality since $\theta_i \geq c_i$. This seems to make the proportional share allocation rule an ideal candidate to obtain budget feasible mechanisms. Indeed, with some minor adjustments, for many problems with functions in the submodular class (e.g. symmetric, additive, and multi-unit demand functions) this general approach works well and produces budget feasible mechanisms with good approximation guarantees (see the following chapter for more details). Furthermore, as we discussed above, the proportional share mechanism is optimal in some cases, and in some restricted environments our characterizations show that this is essentially the *only* budget feasible mechanism (see Section 4.5). The problem however is that this natural approach completely fails as soon as we encounter more involved submodular functions.

The Difficulties

Coverage functions capture many of the difficulties that are associated with designing budget feasible mechanisms for the general submodular case. Recall that a function $f : 2^{[n]} \to \mathbb{R}_+$ is a *coverage* function, if there exist some sets $T_1, \ldots, T_n$ of some universe of elements $\mathcal{U}$, and $f(S) = | \cup_{i \in S} T_i|$. In coverage functions the marginal contribution of an agent is not fixed, but depends on the subset allocated by the algorithm in the previous stages. An agent's marginal contribution therefore depends on its position in the sorting, which introduces several difficulties.

Marginal Contributions are Affected by Costs. When applying the proportional share mechanism in coverage functions, paying agents θ_i as above will be under the budget, as we desire. However, observe that for each agent a_i the payment depends on the marginal contribution f_i, which is determined by a_i's position in the sorting. Thus, in such a case the payments will depend on the agent's declared cost, and therefore cannot induce truthfulness, making the proportional share mechanism hopeless here.

Simple allocation and payment schemes that are independent of the agent's position in the sorting also fail. An approach that may seem natural is to replace marginal contributions with Shapley values [31] since they make the proportional contribution of an allocated agent independent of her position in the sorting. Unfortunately, such an approach cannot approximate better than a factor of $\sqrt{n}$, as we show in the example below. While it is tempting to get rid of the marginal contribution sorting, it is the only known means for obtaining good approximation guarantees for the general submodular case.

Example: Shapley Mechanisms Provide Poor Guarantees. For coverage functions, the proportional share allocation rule can be generalized via Shapley values which are often used in cost sharing (see [31]). Consider a coverage function with subsets (agents) $\{T_1, \ldots, T_n\}$ of a universe U, let $U_j = \cup_{i \in [j]} T_j$ and $\gamma_j(u)$ denote the number of agents in $[j]$ that cover an element $u \in U_j$. In our context, for a set U_j and agent i, the Shapley values are:

$$\xi_{i,j} = \sum_{u \in U_j \cap T_i} \frac{1}{\gamma_j(u)}$$

Note that by definition $|U_j| = \sum_{i \leq j} \xi_{i,j}$. The attractive property of Shapley values is that they make the proportional contribution of an agent independent of the stage in which she was selected by the mechanism. While it seems natural to replace the marginal contributions with Shapley values in the proportional share allocation rule presented above, this results in a poor approximation ratio. Under Shapley values, at every stage, as an item is added to the solution, the proportional share of the rest of the agents can decrease. Individual rationality requires that the mechanism stops at stage k if there is an agent in $\{1 \ldots k-1\}$ whose Shapley value decreases below her cost. To see this can result in a poor approximation ratio consider the following instance. The set $T_1 = \{u_0, u_1\}$ has cost $1 - \epsilon_1$, and the rest of the sets, all with cost $1 - \epsilon_i$, are of the form $T_i = \{u_{m(i)}, u_i\}$, where $m(i) \equiv i \mod 2$, and $\epsilon_1 > \ldots > \epsilon_n > 0$ are small. We set the budget to be $B = n$. For this instance, the mechanism picks T_1 first, and after every odd stage j the value of the solution is $j + 1$, and the elements in T_1 are covered by $j/2$ sets. Therefore at stage j the Shapley value of agent a_1 associated with T_1 is $n/(j^2 + j)$, and thus after the $\sqrt{n}^{th}$ stage no longer exceeds her cost. This gives total value of $\sqrt{n}$, while the optimal solution $\{1 \ldots n\}$ has value of $n + 1$.

Non-monotonicity of the Maximum Operator. Bounded approximation ratios for submodular maximization under a budget constraint depend crucially on the ability to take the maximum between a greedy solution and the item with highest value. In the general case, as well in the case of coverage functions, taking this maximum does not preserve monotonicity: simple examples show that for allocation rules that depend on marginal contribution sorting, by decreasing her cost an agent can *decrease* the value of the allocation.[2]

Example: Applying the MAX Operator Breaks Truthfulness. While a greedy allocation rule that allocates to all items that respect $c_i \leq f_i B/f(S_i)$ is monotone, allocating based on taking the maximum value of this allocated set and another solution is not monotone in the case of general submodular problems. To see this, consider an instance of a coverage function with a universe of elements, partitioned to the following disjoint subsets W, X, Y, Z, with cardinalities $|W| = 7, |X| = 2, |Y| = 2, |Z| = 4$. Let $\{a_0, a_1, a_2\}$ be the set of agents, with $T_0 = W, T_1 = X \cup Y, T_2 = X \cup Z$. Set the budget $B = 1$, and costs $c_0 = \epsilon, c_1 = 7/24$ (a fraction between 1/3 and 1/4) and $c_2 = 1/2$. In this case agent a_1 appears before a_2 in the sorting, both agents are allocated as both satisfy the algorithm's allocation condition $c_i \leq f_i(S_{i-1})/f(S_{i-1} \cup \{i\})$ and $|X \cup W \cup Z| > |W|$. If a_2 declares a lower value, which puts her ahead of a_1, she will no longer be allocated: the marginal contribution of a_1 will be $|Y| = 2$, and since $c_1 > 2/|X \cup Y \cup Z| = 1/4$, only agent a_2 satisfied the condition of the algorithm, and the set covered by the proportional share allocation is therefore $X \cup Z$. Since $|X \cup Z| = 6 \leq |W|$, agent 2 is no longer allocated.

Overview of Our Approach

Our approach is based in three ideas:

- First, we derive an alternative characterization of the threshold payments of the proportional share allocation rule. Since we know that this rule does not work, this may seem futile. We'll show that this characterization plays a significant role in our design.

[2]It is important to emphasize that such examples are not unique to our specific mechanism and not even to coverage functions. This type of problem arises when applying greedy and approximation procedures in other domains as we show in the next chapter.

- Using the above characterization, we show that for any (increasing) submodular function, we can slightly modify the proportional share allocation rule so that its threshold payments are "not too far" from the agents' proportional contributions. This enables us to guarantee that when running the modified version of the proportional share allocation rule *with a constant fraction of the budget*, the threshold payments will be budget feasible.

- Finally, to obtain the approximation guarantee we partition the agents in a manner that allows us to include the variation of the proportional share rule over a *subset of agents* and obtain good approximation guarantees.

Characterizing Threshold Payments

The following definition is key in our characterization.

Definition 10. *The marginal contribution of agent a_i at point j denoted $f_{i(j)}$ is $f(T_{j-1} \cup \{a_i\}) - f(T_{j-1})$ where T_j denotes the subset of the first j agents in the marginal-contribution-per-cost sorting (as in (4.1)) over the subset $\mathcal{N} \setminus \{a_i\}$.*

The intuition behind the payments characterization can be described as follows. Consider running the proportional share mechanism without agent a_i. For the first j agents in the marginal contribution sorting, using the marginal contribution of a_i at point j we can find the maximal cost that agent a_i can declare in order to be allocated instead of the agent in the j^{th} place in the sorting. While these costs may have arbitrary behavior as a function of j, we will show that taking the maximum of these values guarantees payments that support truthfulness. To avoid confusion we use T_j to denote the first j agents according to this sorting, f'_j to denote the marginal contribution of the j^{th} agent in this case, and k' to denote the index of the last agent $j \in \mathcal{N} \setminus \{a_i\}$ that respects:

$$c_j \leq B \cdot \frac{f'_j}{f(T_j)}.$$

For brevity we will write $c_{i(j)} := f_{i(j)} \cdot c_j / f'_j$ and $\rho_{i(j)} := B \cdot f_{i(j)} / f(T_{j-1} \cup \{i\})$.

Lemma 3 (Payments Characterization)**.** *For any submodular function, the threshold payment of the proportional allocation rule is:*

$$\theta_i = \max_{j \in [k'+1]} \Big\{ \min\{c_{i(j)}, \rho_{i(j)}\} \Big\} \quad \forall i \in [n].$$

Proof. To characterize the thereshold payment for agent a_i, relabel the agents according to the marginal-contribution-per-cost sorting over the subset $\mathcal{N} \setminus \{a_i\}$. Consider $\mathcal{A}$ as a sequential allocation rule: at each stage j, the mechanism resorts the remaining agents that have not yet been allocated according to marginal contribution-per-cost sorting, and allocates to the first agent in the sorting if she meets the condition $c_j \leq f'_j \cdot B / f(T_j)$.

For a given stage j in this sequential allocation, we can find the maximal cost agent a_i would have been able declare and be allocated, if she had been considered by the mechanism at this stage: The value $c_{i(j)} = f_{i(j)} \cdot c_j / f'_j$ is the maximal cost a_i can declare which would place her ahead of j in the sorting, and if this cost does not exceed $\rho_{i(j)} = B \cdot f'_{i(j)} / f(T_{j-1} \cup \{a_i\})$, the mechanism would have allocated to agent a_i. Therefore, had a_i appeared at stage j, the minimum between these values is the maximal cost she can declare and be allocated at this stage. Since $f_{i(j)}$ monotonically decreases with j while c_j/f'_j increases, $c_{i(j)}$ may have arbitrary behavior as a function of j. However, as we now show, taking the maximum of these values result in threshold payments.

Let r be the index in $[k'+1]$ for which $\min\{c_{i(j)}, \rho_{i(j)}\}$ is maximal. Declaring a cost below $\theta_i \leq c_{i(r)}$ guarantees a_i to be within the first $r \leq k'+1$ elements in the sorting stage of the mechanism, with $r-1$ items allocated. Since $\theta_i \leq \rho_{i(r)}$, a_i will be allocated.

To see that declaring a higher cost prevents a_i from being allocated, consider first the case where $c_{i(r)} \leq \rho_{i(r)}$. A higher cost places a_i after r in the sorting stage of the mechanism. If the maximum of $c_{i(j)}$ over all $j \in [k'+1]$ is $c_{i(r)}$, reporting a higher cost places a_i after an element which is not allocated and therefore it will not be allocated. Otherwise, if $c_{i(r)} < c_{i(j)}$, for some $j \leq k'+1$, by the maximality of r it must be the case that:

$$\frac{B \cdot f_{i(j)}}{f(T_{j-1} \cup \{a_i\})} = \rho_{i(j)} \leq c_{i(r)} < c_{i(j)}$$

and a_i will not be allocated as a cost above $\rho_{i(j)}$ will not meet the allocation condition.

In the second case when $c_{i(r)} > \rho_{i(r)}$, if r is the index which maximizes $\rho_{i(j)}$ over all indices in $[k'+1]$, reporting a higher cost will not meet the mechanism's allocation condition at each index in $[k'+1]$. Otherwise, if there is some other index $j \in [k'+1]$ for which this maximum is achieved, then:

$$\frac{f_{i(j)} \cdot c_j}{f'_j} = c_{i(j)} \leq \rho_{i(r)} < \rho_{i(j)}$$

and thus declaring a higher cost in this case places a_i after a_j in the sorting, and the mechanism will not consider a_i. □

Lemma 4 (Individual Rationality)**.** *The mechanism that uses the proportional share allocation rule $\mathcal{A}$ and threshold payments θ_i as above is individually rational, i.e., $c_i \leq \theta_i$.*

Proof. Observe that:

(a) $f_{i(j)} \geq f_{i(j+1)} \quad \forall j \in \mathcal{N}$;

(b) $T_j = S_j \quad \forall j < i$;

(c) $f_{i|T_{i-1}} = f_i$.

Since the threshold payment is the maximum over all $\min\{c_{i(j)}, \rho_{i(j)}\}$ in $[k'+1]$, it is enough to show that $c_i \leq \min\{c_{i(j)}, \rho_{i(j)}\}$ for a certain $j \leq k'+1$. Since (b) implies that $i \leq k'+1$ we can consider a_i's replacement a_j which appears in the i^{th} place in the marginal-contribution-per-cost sorting over $\mathcal{N} \setminus \{a_i\}$. Since $i \in [k]$, and due to (b) and (c) above, we have that:

$$c_i \leq \frac{f_i \cdot B}{f(S_{i-1} \cup \{a_i\})} = \frac{f_{i|T_{i-1}} \cdot B}{f(T_{i-1} \cup \{a_i\})} = c_{i(j)}.$$

In the original sorting, a_i appears ahead of a_j (as implied from (b)), and therefore its relative marginal contribution is greater. Thus:

$$c_i \leq \frac{f_{i|S_{i-1}} \cdot c_j}{f_{j|S_{i-1}}} = \frac{f_{i|T_{i-1}} \cdot c_j}{f'_{j|T_{j-1}}} = \rho_{i(j)}.$$

It therefore follows that $c_i \leq \min\{c_{i(j)}, \rho_{i(j)}\} \leq \theta_i$. □

Payment Bounds

The characterization above allows us to include a slightly modified version of the proportional share allocation rule in our mechanism, with threshold payments that are guaranteed to be no more than a constant factor away from agents' proportional contribution. This is a key property which guides the design of our mechanism.

We will run the modified proportional share allocation rule over a subset of the agents $\mathcal{N}_s = \{a_i \in \mathcal{N} : c_i \leq \frac{B}{2}\}$, with a constant fraction of the budget, denoted B'. We discuss the choice of $\mathcal{N}_s$ and B' in the following sections, but for the purpose of showing the payment bounds, we can think of these as any subset of agents and any budget. In this modified version of the proportional share allocation rule, for $a_s^* := \text{argmax}_{a \in \mathcal{N}_s} f(a)$, we sort the agents of $\mathcal{N}_s \setminus \{a_s^*\}$ according to the marginal-contirbution-per-cost order, and allocate to $S = S_k \cup \{a_s^*\}$, where S_k are all k agents in $\mathcal{N}_s$ that respect the condition $c_i \leq f_i \cdot B / f(S_i \cup \{a_s^*\})$. The characterization from above easily extends to this case using $\rho_{i(j)} = f_{i(j)} \cdot B / f(T_{j-1} \cup \{a_i, a_s^*\})$. Under this modification we can show a desirable bound on the threshold payments:

Lemma 5 (Payment Bounds). *For each agent $a_i \in S \setminus \{a_s^*\}$, $\theta_i \leq \left(\frac{4e}{e-1}\right) \frac{f_i \cdot B'}{f(S)}$.*

Proof. For $T_{k'}$ as above, let $S' = T_{k'} \cup \{a_s^*\}$ and let r be the index for which $\theta_i = \min\{c_{i(r)}, \rho_{i(r)}\}$. If $r \leq k'$, observe that the sorting implies $c_r / f'_r \leq c_{k'} / f'_{k'}$ and therefore:

$$\theta_i \leq \frac{f_{i(r)} \cdot c_r}{f'_r} \leq \frac{f_{i(r)} \cdot c_{k'}}{f_{k'}} \leq \frac{f_{i(r)} \cdot B'}{f(S')} \leq \frac{f_i \cdot B'}{f(S')}$$

where the last inequality relies on the observation that $f_{i(j)} \leq f_i$ for every $j \in \mathcal{N}_s \setminus \{a_s^*\}$, which is due to the fact that $T_j = S_j$ for $j \leq i$ and the decreasing marginal utility property of f. In the case where $r = k'+1$:

$$\theta_i \leq \rho_{i(r)} = \frac{f_{i(k'+1)} \cdot B'}{f(S' \cup \{i\})} \leq \frac{f_i \cdot B'}{f(S')}.$$

We therefore see that in both cases $\theta_i \leq f_i \cdot B'/f(S')$. To complete our proof, we will show that $f(S) \leq \Big((4e)/(e-1)\Big) f(S')$.

Let the agents in $\mathcal{N}_s \setminus \{a_i, a_s^*\}$ be sorted as describes above, and let ℓ' be the maximal index s.t. $\sum_{i=1}^{\ell'} c_i \leq B$. For sake of the analysis, consider adding a new agent, a', declares cost $B - \sum_{i=1}^{\ell'} c_i$ and

$$f(a') = \Big(\frac{B - \sum_{i=1}^{\ell'} c_i}{c_{\ell'+1}}\Big) \cdot \Big(f(T_{\ell'+1}) - f(S_{\ell'})\Big)$$

Observe that the weighted marginal contribution of this agent is identical to that of agent $a_{\ell'+1}$ and that the solution $f(S_{\ell'} \cup \{a'\})$ is feasible. Obviously, the optimal solution over all agents in $\mathcal{N}_s \cup \{a'\}$ is an upper bound on the optimal solution over all agents in $\mathcal{N}_s$. Due to the decreasing marginal utilities property, we are guaranteed that the first ℓ' agents selected in both $\mathcal{N}_s$ and $\mathcal{N}_s \cup \{a'\}$ are identical. For notational convenience in the following analysis we will denote agent a' as $a_{\ell'+1}$. For any submodular function f we have that $f(T_{\ell'+1}) \geq (1 - \frac{1}{e}) OPT(\mathcal{N}_s \setminus \{a_s^*\}, B')$, where we use $OPT(X, B')$ to denote the optimal solution over agents in $X \subset \mathcal{N}$ with budget $B' \leq B$. We will use $f(T_{\ell'+1})$ as a benchmark. Since $T_{\ell'+1}$ is feasible:

$$\begin{aligned}
&\sum_{j=k+1}^{\ell'+1} \Big(\frac{c_j}{f(T_j) - f(T_{j-1})}\Big)\Big(f(T_j) - f(T_{j-1})\Big) \\
&\leq \sum_{j=1}^{\ell'+1} \Big(\frac{c_j}{f(T_j) - f(T_{j-1})}\Big)\Big(f(T_j) - f(T_{j-1})\Big) \\
&= \sum_{j=1}^{\ell'+1} c_j = B.
\end{aligned}$$

Since agents are sorted according to their weighted marginal contributions, for every $j \in \{k'+1, \ldots, \ell'+1\}$ we have that:

$$\frac{f(T_j) - f(T_{j-1})}{c_j} \leq \frac{f(T_{k'+1}) - f(T_{k'})}{c_{k'+1}}. \tag{4.2}$$

Putting the above inequalities together we get:

$$\frac{c_{k'+1}}{f(T_{k'+1}) - f(T_{k'})}\Big(\sum_{j=k'+1}^{\ell'+1} f(T_j) - f(T_{j-1})\Big) = \Big(\frac{c_{k'+1}}{f(T_{k'+1}) - f(T_{k'})}\Big) \cdot \Big(f(T_{\ell'+1}) - f(T_{k'+1})\Big) \leq B'.$$

Note that this implies that $f(T_{k'+1}) > f(T_{\ell'+1}) - f(T_{k'})$ as otherwise $c_{k'+1} \leq B'\left(\frac{(f(T_{k'+1}) - f(T_{k'})}{f(T_{k'+1})}\right)$ which contradicts the maximality of k'. Thus, together with submodularity this implies that:

$$f(T_{\ell'+1}) = f(T_{\ell'+1}) - f(T_{k'}) + f(T_{k'}) < f(T_{k'+1}) + f(T_{k'}) \leq 2f(T_{k'}) + f(a_{k'+1}) \leq 3f(S').$$

where the last inequality is due to the fact that a_s^* is included in S'. In conclusion, we get:

$$\begin{aligned} f(S) &\leq OPT(\mathcal{N}_s, B') \\ &\leq OPT(\mathcal{N}_s \setminus \{a_i\}, B') + f(a_i) \\ &\leq \left(\frac{e}{e-1}\right) f(T_{\ell'+1}) + f(a_i) \\ &\leq \left(\frac{3e}{e-1}\right) f(S') + f(a_i) \\ &\leq \left(\frac{4e}{e-1}\right) f(S'). \end{aligned}$$

Since, as we argued above, $\theta_i \leq \frac{f_i \cdot B'}{f(S')}$ this concludes our proof. □

Approximation Guarantee

Now that we've seen that the threshold payments of the modified proportional share allocation rule are bounded, the third step requires arguing about its approximation guarantee. Let $\mathcal{N}_s := \{a_i : c_i \leq B/2\}$. We now show the modified proportional share allocation rule is a constant factor approximation over $\mathcal{N}_s$.

Lemma 6. *Let S be the result of the modified proportional share mechanism as described above computed on $\mathcal{N}_s \setminus \{a_s^*\}$ using budget $B/2\alpha$, for some $\alpha \geq 1$. Then: $\left(\frac{(4\alpha+2)e}{e-1}\right) f(S) \geq OPT(\mathcal{N}_s, B)$.*

Proof. Let the agents in $\mathcal{N}_s \setminus \{a_s^*\}$ be sorted according to the marginal-contribution-per-cost ordering, let k be the largest i s.t. $c_i \leq \frac{B}{2\alpha} \cdot \frac{f_i}{f(S_i)}$, and ℓ be the maximal index s.t. $\sum_{i=1}^{\ell} c_i \leq B$. As in the proof of Lemma 5, for sake of analysis we will compare against the solution produced by $S_{\ell+1}$, and without loss of generality assume it is feasible, and a $(\frac{e}{e-1})$ approximation of $OPT(\mathcal{N}_s \setminus \{a^*\}, B)$.

Similarly to what we have shown in lemma 5, one can show that when applying the proportional share allocation rule with budget $B/2\alpha$ (i.e. $S = \mathcal{A}(c, B/2\alpha, \mathcal{N}_s \setminus \{a_s^*\})$), then:

$$2\alpha f(S_{k+1}) > f(S_{\ell+1}) - f(S_k)$$

and since $a_s^* \in S$ we have that: $f(S_{\ell+1}) < (4\alpha + 1) f(S)$. Therefore:

$$\begin{aligned} OPT(\mathcal{N}_s, B) &\leq OPT(\mathcal{N}_s \setminus \{a^*\}, B) + f(a^*) && (4.3)\\ &\leq \Big(\frac{e}{e-1}\Big) f(S_{\ell+1}) + f(a^*) && (4.4)\\ &\leq \Big(\frac{e}{e-1}\Big)\Big(4\alpha+1\Big) f(S) + f(a^*) && (4.5)\\ &\leq \Big(\frac{e}{e-1}\Big)\Big(4\alpha+2\Big) f(S) && (4.6) \end{aligned}$$

which implies our desired bound. □

Main Result

Given all the above, we can now prove our main theorem.

Theorem 4.4.1. *For any submodular utility function there exists a constant factor approximation randomized mechanism in the value query model which is budget feasible and universally truthful. Furthermore, no budget feasible mechanism can do better than $2-\epsilon$, for any fixed $\epsilon > 0$.*

In our discussion we will use $\alpha = \frac{4e}{e-1}$. Our analysis shows our mechanism guarantees an approximation ratio of $\left(\frac{(16\alpha+8)e}{e-1}\right) \approx 173.6$ in expectation. It is possible that tighter analysis can show the mechanism does better.

Proof. Consider the following mechanism:

A Randomized Mechanism for Nondecreasing Submodular Functions

```
1. Set $\mathcal{N}_s = \{a_i \in \mathcal{N} : c_i \leq \frac{B}{2}\}$, $a_s^* \in \text{argmax}_{a \in \mathcal{N}_s} f(a), S = \{a_s^*\}, a_i \in \text{argmax}_{a \in \mathcal{N}_s \setminus \{a^*\}} \frac{f(a)}{c_i}$
2. While $c_i \leq \frac{B}{2\alpha} \cdot \left(\frac{f(S \cup \{a_i\}) - f(S)}{f(S \cup \{a_i\})}\right)$:
    a. Add $a_i$ to $S$
    b. Set $a_i \in \text{argmax}_{a \in \mathcal{N}_s \setminus \{a^*\}} \frac{f(S \cup \{a\}) - f(S)}{c_j}$

Output: Choose u.a.r between $S$ and $\text{argmax}_{a \in \mathcal{N} \setminus \mathcal{N}_s} f(a)$
```

If S is chosen then the payment is $B/2$ to a_s^* and $\hat{\theta}_i = \min\{\theta_i, B/2\}$ for $a_i \neq a_s^*$, where θ_i is the threshold payment described in Lemma 3, when the modified proportional share rule is used: $\mathcal{A}(\mathcal{N}_s \setminus \{a_s^*\}, B/2\alpha)$. Observe the budget used here is exactly half of the constant from the bound in Lemma 5.

First, observe that the mechanism is monotone. For agents in $\mathcal{N} \setminus \mathcal{N}_s$, the allocation is oblivious to their cost. For agents in $\mathcal{N}_s$, a_s^* will be in S for any cost she declares that is below $B/2$, and it is easy to verify that each other agent in S can reduce her cost and continue to be allocated by the modified proportinal share allocation. Therefore in each realization of the random coin, the mechanism is monotone.

Regarding threshold payments, in case $a \in \text{argmax}_{a \in \mathcal{N} \setminus \mathcal{N}_s} f(a)$ is chosen, then clearly, B is her threshold payment, and the solution is budget feasible. If S is allocated, from the characterization lemma and the fact that $\mathcal{N}_s$ consists only of agents with cost less than $B/2$, we know that $\hat{\theta}$ as described above are clearly the threshold payments. Since the modified proportional share allocation rule uses $B' = B/2\alpha$ as its budget, from Lemma 5, we can conclude that:

$$\sum_{a_i \in S} \hat{\theta}_i \leq \frac{B}{2} + \alpha \sum_{a_i \in S \setminus \{a_s^*\}} \frac{f_i \cdot B'}{f(S)} \leq B$$

and the mechanism is therefore truthful and budget feasible. Individual rationality and monotonicity were discussed above, and the lower bound as described in the following section applies here as well.

Finally, let $\beta = \left(\frac{(4\alpha+2)e}{e-1}\right)$ and let $a^* = \text{argmax}_{\mathcal{N} \mathcal{N}_s} f(a)$. Observe that due to the budget constraint, the optimal solution cannot include more than one item from $\mathcal{N} \setminus \mathcal{N}_s$, and its value is no greater than that of a^*. This, together with Lemma 6 implies:

$$\begin{aligned} OPT(B, \mathcal{N}) \leq \; & OPT(B, \mathcal{N}_s) + f(a^*) \\ & \leq \beta f(S) + f(a^*) \\ & \leq 2 \max\{\beta f(S), f(a^*)\} \end{aligned} \tag{4.7}$$

and therefore gives us our desired approximation ratio. □

4.5 Discussion

The space of budget feasible mechanisms appears quite rich and invites further investigation. The richness of the submodular class implies there are many problems for which better approximation ratios are achievable. In the following chapter we present several important subclass of submodular functions and show improved approximation guarantees. We conclude this chapter with a brief discussion about the model.

First, we ask whether problems studied in the frugality framework can have good budget feasible approximations. We show a simple lower bound which implies that budget feasibility and frugality are complementary models: for superadditive objectives where we aim to optimize the payments as long as the outcome meets some threshold, the frugality model is most appropriate. In cases where we care about the quality of the outcome, as long as the payments meet some threshold, budget feasibility is the right model.

The second question we address is concerned with the type of mechanisms we can expect in the budget feasibility framework. We give a simple characterization under restrictive conditions, that gives some intuition for the structure of budget feasible mechanisms.

The Yin of Budget Feasibility and the Yang of Frugality

Since procurement has been studied in the model of frugality in mechanism design, it is perhaps most natural to see whether the "hiring a team of agents" problems studied in the frugality framework [4, 58, 32, 15] have good budget feasible mechanisms. In these problems there is a set of feasible outcomes (e.g. all possible spanning trees or all source-destination paths in a graph) and the goal is to design a mechanism that yields a feasible solution. In the literature these problems have been studied with the goal of designing mechanisms which yield minimal payments (frugal), according to various benchmarks.

In our notation, such problems can be written as having a function $f(S) = 1$ if $S \in \mathcal{F}$, and 0 otherwise, where $\mathcal{F}$ is the set of all feasible outcomes. Call such a problem *nontrivial* if all solutions in $\mathcal{F}$ contain more than one element.

Theorem 4.5.1. *There is no budget feasible mechanism with a bounded approximation ratio for any nontrivial "hiring a team of agents" problem.*

Proof. Assume for purpose of contradiction, there exists a a bounded approximation ratio mechanism $\mathcal{A}$ which is truthful and budget feasible. Let S be a feasible solution, and consider the bid profile in which all agents in S declare positive cost $\epsilon < B/|S|$, and all other agents declare B. Since the problem is nontrivial, the minimal cost of a feasible solution different than S (if it exists) exceeds the budget. Since A guarantees a bounded approximation ratio, it must allocate to S. For agent $a_i \in S$, consider the cost vector (c'_i, c_{-i}) with $c'_i = B - \epsilon(|S| - 1)$. Observe that S remains a cost-feasible solution, and the only one in $\mathcal{F}$, and therefore $f(c'_i, c_{-i}) = S$, which implies the threshold payment of agent a_i is at least c'_i. Since this holds for all agents in S, it contradicts budget feasibility. □

Characterizing Budget Feasible Mechanisms in Restricted Settings

The characterization of budget feasible mechanisms with good approximation ratios naturally depends on the environment in which the mechanisms are implemented, or more concretely, the function we aim to optimize. In characterizations, we often introduce additional restrictions on our mechanisms (e.g. maximal-in-range) which we then use as guidelines in their design or for complete characterization.

We now consider mechanisms that respect the two additional conditions of anonymity [6] and weak stability (similar to [21]). Informally, a mechanism is weakly stable if an agent doesn't hurt the rest when reducing her cost, and anonymous if its allocation rule does not depend on the agents' identities.

Definition 11. *An allocation rule $\mathcal{A}$ satisfies weak stability if for every $a_i, a_j \in \mathcal{A}(c_i, c_{-i})$, $c_i' \leq c_i$ implies $a_j \in \mathcal{A}(c_i', c_{-i})$.*

Definition 12. *An allocation rule $\mathcal{A}$ satisfies anonymity if $a_i \in \mathcal{A}(c_i, c_j, c_{-ij})$ implies $a_j \in \mathcal{A}(c_i', c_j', c_{-ij})$ when $c_i' = c_j, c_j' = c_i$.*

Note that for the class of symmetric submodular functions, the proportional share mechanism respects these conditions. Furthermore, it seems quite reasonable that mechanisms with good approximation properties satisfy these conditions. The theorem below tells us that for allocation functions that carry these properties, the proportional share mechanism is essentially the only mechanism we can expect.

Theorem 4.5.2. *Let $\mathcal{A}$ be a budget feasible mechanism that is anonymous and weakly stable, and let $S = \mathcal{A}(c)$ for some bid profile c. Then, for all $a_i \in S$ it must be that $c_i \leq B/|S|$.*

Proof. Assume for purpose of contradiction that there is a bid profile $c = (c_1, \ldots, c_n)$, s.t. $f(c) = S$ and there is some $a_i \in S$ for which $c_i > B/|S|$. Let c' be the bid profile in which all agents in $S \setminus \{a_i\}$ bid $c_{min} = \min_{j \in S} c_j$, and the rest bid as in c. Since $\mathcal{A}$ is weakly stable, we have that $S \subseteq \mathcal{A}(c')$. Let c'' the bid profile where a_i bids c_{min} as well, and the rest of the agents bid as in c'. From monotonicity we have that a_i is allocated, and again $S \subseteq \mathcal{A}(c'')$. We now claim that under the profile c'', the threshold price for each agent in S is at least $c_i > B/|S|$. To see this, observe that a_i's threshold price must be at least c_i, since $a_i \in \mathcal{A}(c')$. Since $\mathcal{A}$ is anonymous, and all agents in S declare the same price, the threshold price for each agent in S must also be at least c_i. Thus, payments to agents in S exceed the budget, contradicting budget feasibility. $\square$

Chapter 5

Budget Feasible Mechanisms II: Adventures in Approximation

5.1 Introduction

In the previous chapter we introduced the budget feasibility model, and showed that for any increasing submodular function there exists a randomized budget feasible mechanism, which in expectation obtains a constant factor approximation. In this chapter we discuss other important classes of objective functions in the budget feasibility model. When introducing the model we showed that even for very simple *superadditive* objectives (that is, objectives in which $f(S \cup T) > f(S) + f(T)$ for some sets S, T) budget feasible mechanisms with bounded approximation ratios do not exist. In view of this, in this chapter we consider objectives that are *complement-free*, or *subadditive*, valuations obeying $f(S \cup T) \leq f(S) + f(T)$. We seek approximation guarantees in three different categories:

- **Subclasses of submodular functions.** The main result from the previous chapter holds for the general class of increasing submodular functions. For stricter classes (e.g. additive functions) many of the problems of submodular functions can be avoided and allow for *deterministic* mechanisms as well as better approximation ratios.
- **Non-increasing functions.** For cases in which $S \subseteq T$ does not necessarily imply $f(S) \leq f(T)$ we say that f is non-increasing. The previous chapter discusses *increasing* functions, and we wish to understand whether budget feasible mechanisms exist for the non-increasing case as well.
- **General Subadditive functions.** The class of subadditive functions is a substantial generalization of submodular functions. Since simple superadditive objectives require huge overpayments in order to be approximated within a reasonable factor, subadditive functions are the most general class in which budget feasibility may exist. Thus, understanding whether budget feasible mechanisms are obtainable in this class is a fundamental open question.

The Results in the Chapter

We discuss five different classes of utility functions in this chapter. We first show that for additive, multi-unit demand, and coverage – all of which are subclasses of increasing submodular functions – there are deterministic mechanisms with better approximation guarantees than that of the general result presented in the previous chapter. We then discuss the class of cut functions, which are a nontrivial case of utility functions that can decrease in value as items are added. The results from the previous chapter do not apply to this class of utilities since cut functions are non-increasing. Still, for such functions we give a deterministic constant factor approximation mechanism. Finally, for complement-free (subadditive) objectives we first give a randomized $O(\log^2 n)$ approximation mechanism that is universally truthful and budget feasible. We derandomize this mechanism and present a deterministic mechanism for this domain with an $O(\log^3 n)$ approximation guarantee.[1]

The deterministic mechanisms in this chapter are based on their randomized analogues. In all cases, preserving truthfulness in the derandomization is nontrivial and requires constructing "monotone estimators" that are crucial for obtaining bounded approximation guarantees. Each construction uses a different technique, though their underlying principle is similar. We discuss these issues in the appropriate sections.

5.2 Additive Functions

Recall that an additive function is simply the sum of individual contributions of items. Formally, a function $f : 2^{[n]} \to \mathbb{R}_+$ is *additive* if for all $S \subseteq [n]$ we have that $f(S) = \sum_{i \in S} f(i)$.

To simplify notation, we will use v_i to denote the value of each item in $a_i \in \mathcal{N}$, i.e. $v_i = f(a_i)$. Observe that without strategic considerations, the additive case is the well-known knapsack problem: Given a budget B and a set of items $\mathcal{N}$, each with nonnegative cost c_i and value v_i, find a subset $S \in \text{argmax}_{T:\sum_{i \in T} c_i \leq B} \sum_{i \in T} v_i$. For the knapsack optimization problem, there is a *Fully Polynomial-Time Approximation Scheme* (FPTAS).From Proposition 4.3.2 we know we cannot get an approximation better than 2, even in simpler cases where all $v_i = 1, \forall i \in [n]$. In this section we show a variation of the proportional share mechanism which was the basis for the mechanism for submodular functions from the previous chapter, to obtain a deterministic budget feasible mechanism for additive functions which is a 4-approximation.

A Budget Feasible Mechanism for Additive Functions

The mechanism is based on the proportional share rule presented in the previous chapter. We use a^* to denote the agent with the largest value, i.e. $a^* \in \text{argmax}_{a \in \mathcal{N}} f(a)$.

[1]Since subadditive functions may require representation that is exponential in the number of agents, we assume we are given access to a *demand* oracle. The weaker value query oracles result in inapproximability, a stronger oracle model such as the one used here is required. We discuss this point at detail in the appropriate section.

A Deterministic Budget Feasible Mechanism for Additive Functions

1. Reorder agents in $\mathcal{N} \setminus \{a^*\}$ s.t. $\frac{v_1}{c_1} \geq \ldots \geq \frac{v_{n-1}}{c_{n-1}}$
2. Let S be the set of all agents in $\mathcal{N} \setminus \{a^*\}$ that respect $c_i \leq B\left(\frac{v_i}{\sum_{j \leq i} v_j}\right)$

Output: If $\sum_{i \in S} v_i \geq f(a^*)$ then output S, otherwise output agent a^*.

Let k be the maximal index i that respects $c_i \leq B\left(\frac{v_i}{\sum_{j \leq i} v_j}\right)$. If a^* is allocated her payment is B and if $S = [k]$ is allocated the payment is to each $a_i \in S$ is:

$$\theta_i = \min\left\{\frac{v_i \cdot c_{k+1}}{v_{k+1}}, \frac{v_i \cdot B}{\sum_{j \leq k} v_j}\right\}$$

The main insight here is that unlike the general submodular case from the previous chapter, the agents' marginal contributions are fixed values that are independent of the order in which they are considered by the mechanism.

Lemma 7. *The mechanism is truthful.*

Proof. This is a single parameter setting, and thus according to Myerson's characterization it is enough to show that the allocation rule is monotone and that the payments are threshold payments.

To see the allocation is monotone, let the agents bids be $c = (c_1, \ldots, c_{n-1})$ and c^* for a^*, and let $S = [k]$ be the set that is chosen by the greedy procedure. In case $f(a^*) > \sum_{i=1}^{k} v_i$, to see that the allocation is monotone consider the bid profile where a^* declares her cost to be $c' < c^*$. The greedy allocation over $[n-1]$ remains the same, thus S remains the same, and since $v^* > \sum_{i \in [k]} v_i$ we have that a^* will be allocated under this bid profile as well, which implies monotonicity. This argument holds for any bid below B, and thus B is the threshold payment for a^*.

In case $S = [k]$ is allocated, consider an agent $a_i \in S$. First, note that the allocation is monotone: declaring a cost $c'_i < c_i$ will place a_i in position $i' \leq i$, and since

$$c'_i < c_i \leq B \cdot \frac{v_i}{\sum_{j \leq i} v_j} \leq B \cdot \frac{v_i}{\sum_{j \leq i'} v_j}$$

a_i will be selected in the proportional share step. Since the rest of the agents values do not change, all other agents in $S \setminus \{a_i\}$ will be selected in the proportional share rule as well and the sum of their values will be greater than $f(a^*)$, and they will be allocated. Thus, the mechanism is monotone. To see that θ_i is the threshold payment, note that due to the sorting of the agents we have that:

$$c_i \leq v_i \cdot \Big(\frac{c_{k+1}}{v_{k+1}}\Big) \tag{5.1}$$

and due to the proportional share allocation condition we have that:

$$c_i \leq B \cdot \Big(\frac{v_i}{\sum_{j \leq i} v_j}\Big) \leq B \cdot \Big(\frac{v_i}{\sum_{j \leq k} v_j}\Big) \tag{5.2}$$

Thus declaring a cost below θ_i is an upper bound on the threshold price. Observe that declaring a value above this value will deny the agent from being allocated. When $\theta_i = v_i \cdot c_{k+1}/v_{k+1}$ declaring any cost $c'_i > \theta_i$ places i after agent $k+1$ and since $c_{k+1}/v_{k+1} > B/\sum_{j \leq k+1} v_j$, it will not be allocated. When $\theta_i = v_i \cdot B/\sum_{j \leq k} v_j$ declaring a higher cost will place at least $k-1$ items before i, and it will not be allocated. □

Lemma 8. *The mechanism is individually rational and budget feasible.*

Proof. If a^* is allocated, she is paid B which is trivially individually rational and budget feasible. Otherwise, $S = [k]$ is allocated, to see that the mechanism is individually rational and budget feasible, observe that:

$$\frac{c_1}{v_1} \leq \ldots \leq \frac{c_k}{v_k} \leq \min\Big\{\frac{c_{k+1}}{v_{k+1}}, \frac{B \cdot v_k}{\sum_{i \leq k} v_i}\Big\} \leq \frac{v_k \cdot B}{\sum_{i \leq k} v_i}.$$

The second inequality from the right implies individual rationality and the rightmost inequality implies budget feasibility. □

Lemma 9. *The mechanism is a 4-approximation.*

Proof. Recall that we use k to denote the maximal index i for which $c_i \leq B \cdot \frac{v_i}{\sum_{j \leq i} v_j}$ and we will use ℓ to denote the maximal index for which $\sum_{i \leq \ell} c_i \leq B$. Similar to the proofs from the previous chapter, observe that the optimal integral solution over $\mathcal{N} \setminus \{a^*\}$ is bounded from above by the optimal *fractional* solution. Therefore:

$$OPT(\mathcal{N} \setminus \{a^*\}, B) \leq \sum_{i=1}^{\ell} v_i + \Big(\frac{B - \sum_{i=1}^{\ell} c_i}{c_{\ell+1}}\Big) \cdot v_{\ell+1}.$$

For sake of the analysis we will assume $v_{\ell+1} = \Big(\frac{B - \sum_{i=1}^{\ell} c_i}{c_{\ell+1}}\Big) \cdot v_{\ell+1}$ and $c_{\ell+1} = B - \sum_{i=1}^{\ell} c_i$. First, we will show that $\sum_{i=1}^{k} v_i > \sum_{i=k+2}^{\ell+1} v_i$. Note that:

$$\sum_{i=k+1}^{\ell+1} v_i = \sum_{i=k+1}^{\ell+1} \big(\frac{c_i}{v_i} \cdot v_i\big) \leq \frac{c_{k+1}}{v_{k+1}} \sum_{i=k+1}^{\ell+1} v_i$$

and we can therefore conclude that $c_{k+1} \leq B \cdot \frac{v_{k+1}}{\sum_{k+1}^{\ell+1} v_i}$. Due to the maximality of k this necessarily implies:

$$\sum_{i=1}^{k+1} v_i > \sum_{k+1}^{\ell+1} v_i$$

and we therefore get that: $\sum_{i=1}^{k} v_i > \sum_{i=k+2}^{\ell+1} v_i$. In conclusion, we get:

$$\begin{aligned} OPT &\leq OPT(\mathcal{N} \setminus \{a^*\}, B) + f(a^*) \\ &\leq \sum_{i=1}^{\ell+1} v_i + f(a^*) \\ &= \sum_{i=1}^{k} v_i + v_{k+1} + \sum_{i=k+2}^{\ell+1} v_i + f(a^*) \\ &< 2\sum_{i=1}^{k} v_i + 2f(a^*) \qquad (5.3) \\ &\leq 4 \max\Big\{ \sum_{i=1}^{k} v_i, f(a^*) \Big\} \end{aligned}$$

Thus choosing the maximum between $\sum_{i=1}^{k} v_i$ and $f(a^*)$ is a 4-approximation. □

Together, the above lemmas give us the following theorem.

Theorem 5.2.1. *For additive functions there is a deterministic truthful 4-approximation mechanism which is budget feasible.*

5.3 Multi-Unit Demand Functions

In this section we discuss Multi-unit demand (called OXS in [37]) functions which can be defined as a sum of unit demand functions. Recall that a function $f : 2^{[m]} \to \mathbb{R}$ is called *unit-demand* if for all $T \subseteq [m]$ there exist weights $w_1, \ldots, w_m$ s.t. $f(T) = \max_{i \in T} w_i$.

Definition 13. *A function $f : 2^{[n]} \to \mathbb{R}_+$ is called* multi-unit demand *if there exists some unit-demand functions $f_1, \ldots, f_m$ s.t. $f(S) = \sum_{i=1}^{m} f_i(S)$.*

The multi-unit demand class of functions and is known to be submodular [37] and can be thought of in terms of a one-sided matching on a weighted graph.

Definition 14. *Let $G = (V \cup U, E)$ be a bipartite graph. A one sided matching is a set of edges in E where each edge has only one end point in U (but may have many end points in V). Given an edge-weighted graph, the weight of a one-sided matching is the sum of the weights of the edges in the matching, and the optimal one-sided matching is the one-sided matching with the highest weight.*

For multi-unit demand functions, the optimization problem can be stated as a matching problem: Given a budget B, bipartite graph $G = (V \cup U, E)$, where each edge $e \in E$ has nonnegative cost c_e and weight w_e, find a one-sided matching S in the set of all one-sided matchings $\mathcal{F}$ s.t. $S \in \text{argmax}_{T \in \mathcal{F}: \sum_{e \in T} c_e \leq B} \sum_{i \in T} w_e$.

The Mechanism for Multi-Unit Demand

We will use the same mechanism as in the additive case, with a minor adjustment. We will use the broader definition of density sorting, and sort agents according to their marginal contribution (rather than simply the values as in the additive case). More specifically, at each stage i we select the agent (edge) $e_j = (u_j, v_j)$ for which $c_j / w_{j(i)}$ is minimal, where $w_{j(i)} = w_j$ if the set of previously selected agents $e_1, \ldots, e_{i-1}$ do not include u_j and 0 otherwise.

Observe that unlike the additive case, the marginal contribution of an agent is affected by the order in which it is considered in the mechanism. Fortunately, the structure of multi-unit demand functions is simple, and we can make a minor adjustment to the proportional share mechanism from the additive case.

Let $a^* = \text{argmax}_{a \in \mathcal{N}} w_a$ and consider the agents in $\mathcal{N} \setminus \{a^*\}$ sorted according to the density rule where:

$$\frac{c_1}{w_1} \leq \frac{c_2}{w_{2(1)}} \leq \ldots \leq \frac{c_n}{w_{n(n-1)}}$$

and let k be the maximal index i that respects the proportional share rule:

$$c_i \leq B \cdot \Big(\frac{w_{i(i-1)}}{\sum_{j \leq i} w_{j(j-1)}} \Big).$$

This gives us a modified proportional share allocation rule, similar to the additive case. To define the threshold payments, for each edge e, let $\bar{e}$ be the edge that minimizes c_i / w_i over all edges i that share a vertex with e, and let $c_{e(k+1)}$ be the minimum of $c_{\bar{e}} / w_{\bar{e}}$ and $c_{k+1} / w_{(k+1)(k)}$. The threshold payment for the modified proportional share allocation rule would be:

$$\theta_e = \min \Big\{ \frac{w_e \cdot B}{\sum_{j \leq k} w_{j(j-1)}}, c_{e(k+1) \cdot w_e} \Big\}.$$

The same reasoning as in additive case can be applied here to show these are indeed the threshold payments, and that taking the maximum between the greedy allocation and a^* does not break truthfulness. The approximation guarantee can also be derived using similar ideas as in the additive case. Since multi-unit demand are a special case of submodular functions, using similar arguments to those in the previous chapter we can show that:

$$3max\{\sum_{i\leq k} w_i + w_{k+1}, f(a^*)\}2\sum_{i\leq k} w_i + w_{k+1} \geq \Big(1-\frac{1}{e}\Big)OPT_{n-1}$$

where OPT_{n-1} denotes the value of the optimal solution over $\mathcal{N} \setminus \{a^*\}$. This implies:

$$OPT \leq OPT(\mathcal{N}\backslash\{a^*\}, B)+f(a^*) \leq \Big(\frac{3e}{e-1}\Big)\max\Big\{\sum_{i=1}^{k} w_i, w_{k+1}\Big\}+f(a^*) \leq \Big(\frac{4e}{e-1}\Big)\max\Big\{\sum_{i=1}^{k} w_i, f(a^*)\Big\}$$

The maximum between a^* and S gives a $(\frac{4e}{e-1})$-approximation guarantee.

Theorem 5.3.1. *For multi-unit demand functions there is a deterministic truthful* $\Big(\frac{4e-1}{e-1}\Big)$-*approximation mechanism which is budget feasible.*

5.4 Coverage Functions

Coverage functions are an important class of submodular functions. They often capture the main intricacies of submoudlar functions, yet they can be succinctly described. Recall that a function $f : 2^{[n]} \to \mathbb{R}_+$ is a *coverage* function, if there exist some sets $T_1, \dots, T_n$ of some universe of elements $\mathcal{U}$, and $f(S) = |\cup_{i\in S} T_i|$.

For coverage functions, the optimization problem is the budgeted Max-t-Cover problem: Given a budget B and family of sets $T_1, \dots, T_n$ over some universe $\mathcal{U}$, where each set has some cost $c_i \in \mathbb{R}_+$, find a family $S \in \text{argmax}_{T:\sum_{i\in T} c_i \leq B} |\cup_{i\in T} T_i|$.

Mechanism for Coverage Functions

In the previous chapter we presented a randomized mechanism for general submodular functions. In this section we introduce a derandomization technique which is applicable to coverage functions, as well as other functions as we shall see in the following sections.

Recall that the mechanism for the general submodular case randomizes between a^* and the set that is selected by the modified proportional share allocation rule, S. If it were not for incentive constraints, one could simply compare $f(S)$ and $f(a^*)$ and by selecting the solution that yields the higher value, guarantee a constant-factor approximation. In our case however, as shown in example 4.4 this process would violate incentive compatibility. This happens since there are cases where by declaring a lower cost, an agent changes the allocation of the mechanism which may reduce the value of the proportional share solution. That is, letting $\mathcal{A}$ denote the proportional share allocation, there may be cases where $f(\mathcal{A}(c_i, c_{i-1})) > f(a^*)$ but $f(\mathcal{A}(c_i', c_{i-1})) < f(a^*)$ when $c_i' < c_i$. To address this, we develop the following methodology inspired by [8]. We will compare between a^* and a solution to a nonlinear relaxation program that doesn't suffer from this property while the value of its solution is provably "close" to that of the proportional share mechanism. The main idea is simple. Comparing a^* to the solution of the relaxation doesn't break incentive compatibility, and since the solution of the relaxation is close to that of $f(S)$, the approximation ratio doesn't suffer by much. First, observe that the optimization problem can be reformulated as the following integer program:

$$\begin{aligned}
\max \quad & \textstyle\sum_{j=1}^{m} z_j & j \in U \quad & (5.4)\\
& \textstyle\sum_{i \in C_j} x_i \geq z_j, & j \in U, \quad & (5.5)\\
& \textstyle\sum_{i=1}^{n} c_i x_i \leq B, & & (5.6)\\
& 0 \leq x_i, z_j \leq 1 & i \in \mathcal{N}, j \in \mathcal{U}, \quad & (5.7)\\
& x_i \in \{0,1\} & i \in \mathcal{N} \quad & (5.8)
\end{aligned}$$

where z_j are variables representing the agents we wish to cover, x_i are the variables representing the agents we can select, and C_j denotes the subset of agents that cover an agent $a_j \in \mathcal{N}$. For a relaxation we will use:

$$\begin{aligned}
\max \quad & \textstyle\sum_{j=1}^{m} \min\left\{1, \sum_{i \in C_j} x_i\right\} & j \in U \quad & (5.9)\\
& \textstyle\sum_{i \in C_j} x_i \geq z_j, & j \in U, \quad & (5.10)\\
& \textstyle\sum_{i=1}^{n} c_i x_i \leq B, & & (5.11)\\
& 0 \leq x_i, z_j \leq 1 & i \in \mathcal{N}, j \in \mathcal{U}, \quad & (5.12)\\
& x_i \in [0,1] & i \in \mathcal{N} \quad & (5.13)
\end{aligned}$$

The optimal solution to this relaxation, denoted x^* is computable in polynomial time and is a relaxation of the integer program in the sense that for all $x \in \{0,1\}$ it is equivalent to the integer program. In our mechanism we will compare a^* and this relaxation to circumvent the problem with incentive compatibility. We denote this relaxation as L, and use $L(x)$ to denote a solution that respects the constraints above where $x \in [0,1]^n$. We will show that the solution of $L(x)$ is "close" to $f(S)$, which guarantees that comparing a^* to $L(x)$ rather than $f(S)$ does not hurt the solution by more than a constant factor. We will use $L(x^*)$ to denote the value of the optimal solution to the relaxation over the subset $\mathcal{N}_s = \{a_i \in \mathcal{N} \mid c_i \leq B/2\mathcal{N}$ (i.e., we replace $\mathcal{N}$ with $\mathcal{N}_s$ in conditions (5.12) and (5.13) above).

The mechanism for coverage is a derandomized version of the mechanism for general submodular functions from the previous chapter. First, the mechanism uses the modified proportional allocation rule $\mathcal{A}$ from the previous chapter over the subset $\mathcal{N}_s$ and finds a set of agents S (recall that $a_s^* \in \text{argmax}_{a \in \mathcal{N}_s} f(a)$ is always included in S). Rather than randomizing between $a^* \in \text{argmax}_{a \in \mathcal{N} \setminus \mathcal{N}_s} f(s)$ and S, the mechanism compares $f(a^*)$ with $L(x^*)$. If $L(x^*)$ is greater it allocates to S, and otherwise allocates to a^*. First, we show that such a comparison preserves monotonicity.

Lemma 10. *The mechanism described above is monotone.*

Proof. To see that the mechanism is indeed monotone assume for purpose of contradiction that for a given cost profile c there is an agent $a_i \in S$ with declared cost c_i, that is not allocated when declaring $c_i' \leq c_i$ while the rest of the agents declare the same cost as in c. If $a_i \neq a^*$ then it has been selected by the modified proportional share allocation rule (denoted $\mathcal{A}$) and $L(x^*) > f(a^*)$. All agents in S will also be selected by $\mathcal{A}$ when declaring a lower cost. If $a_i = \text{argmax}_{a \in \{a_i : c_i \leq B/2\}} f(a)$ it is always selected by $\mathcal{A}$. Otherwise, by decreasing her cost the agent will move ahead in the sorting to place $j \leq i$, and due to the decreasing marginal contribution property her marginal contribution will only increase. Thus, $f(S_{j-1} \cup \{a_i\}) - f(S_{j-1}) \geq f(S_{i-1} \cup \{a_i\}) - f(S_{i-1})$ and since f is an increasing function $f(S_{j-1} \cup \{a_i\}) \leq f(S_i)$. This implies:

$$c_i' < c_i \leq \frac{B}{2\alpha}\Big(\frac{f(S_i) - f(S_{i-1})}{f(S_i)}\Big) \leq \frac{B}{2\alpha}\Big(\frac{f(S_{j-1} \cup \{a_i\}) - f(S_{j-1})}{f(S_{j-1}) \cup \{a_i\}}\Big)$$

and thus a_i will meet the proportional share allocation rule[2]. Since $L(x^*)$ is the value of the *optimal* fractional solution, it can only increase when the costs are reduced, and therefore its value will be larger than that of a^* in this case as well. In case $a_i = a^*$, it is allocated only if $L(x^*) < f(a^*)$. Note that her cost does not change the value of $f(a^*)$, nor that of $L(x^*)$ which computes its solution over $\mathcal{N} \setminus \{a^*\}$. Thus, $f(a^*)$ remains larger and a^* is allocated. In summary, in all cases we get a contradiction. $\square$

The argument for showing that the mechanism has a constant factor guarantee depends on showing that $L(x^*)$ and the value of the proportional share solution $f(S)$ are close. Consider the nonlinear program relaxation:

$$\begin{aligned}
\max \quad & \textstyle\sum_{j=1}^{m}(1 - \Pi_{i \in C_j}(1 - x_i)) & j \in U \qquad & (5.14)\\
& \textstyle\sum_{i \in C_j} x_i \geq z_j, & j \in U, \qquad & (5.15)\\
& \textstyle\sum_{i=1}^{n} c_i x_i \leq \frac{B}{2}, & & (5.16)\\
& 0 \leq x_i, z_j \leq 1 & i \in \mathcal{N}, j \in \mathcal{U}, \qquad & (5.17)\\
& x_i \in [0,1] & i \in \mathcal{N} \qquad & (5.18)
\end{aligned}$$

Let F denote the above relaxation. Note that both $F(x)$ and $L(x)$ identify with our integer program over integral solutions. The following lemma can be shown directly using the pipage rounding technique in [1].

Lemma 11 ([1]). *Let x^* be the optimal solution for L. Then, there exists an integral solution $\bar{x}$ s.t. $L(x^*) \leq (\frac{2e}{e-1})F(\bar{x})$.*

[2]Recall that α in the above inequality is the constant we use to "shrink" the budget in the general submodular case to achieve budget feasibility.

Using the above lemma, we can now conclude show that the mechanism which uses the comparison discussed above is a constant factor approximation of the optimal soltion. Recall that the optimal *fractional* solution $L(x^*)$ is computed over $\mathcal{N}_s$. First, in Lemma 6 from the previous chapter we showed that there is a constant $\beta = \left(\frac{(4\alpha+2)e}{e-1}\right)$ s.t. $OPT(\mathcal{N}_s, B) < \beta f(S)$. Since F is equivalent to our integer program over integral solutions, then $OPT(\mathcal{N}_s, B)$ is an upper bound on any of its integral solutions. Together with Lemma 11 these two bounds imply:

$$L(x^*) \leq \Big(\frac{2e}{e-1}\Big) OPT(\mathcal{N}_s, B) \ < \beta\Big(\frac{2e}{e-1}\Big) f(S)$$

If $L(x^*) \geq f(a^*)$, from the above inequality we get:

$$OPT \leq OPT(\mathcal{N}_s, B) + f(a^*) \leq OPT(\mathcal{N}_s, B) + L(x^*) \leq \beta\Big(\frac{2e}{e-1} + 1\Big) \cdot f(S)$$

and we indeed have a constant factor approximation. Otherwise, if $L(x^*) < f(a^*)$, then since $L(x^*)$ is an optimal fractional solution over $\mathcal{N}_s$ it is an upper bound on $OPT(\mathcal{N}_s, B)$. Thus in this case since $OPT \leq OPT(\mathcal{N}_s, B) + f(a^*) \leq 2f(a^*)$ the solution is a constant factor approximation as well.

Theorem 5.4.1. *For coverage functions there is a deterministic truthful budget feasible mechanism that obtains a constant factor approximation.*

5.5 Cut Functions

The utility functions we discussed in this chapter were increasing: functions for which $S \subset T$ implies $f(S) \leq f(T)$. However, there are utility functions where this condition does not necessarily apply. In cases where $S \subseteq T$ does not necessarily imply $f(S) \leq f(T)$, we refer to the utility function as *non-increasing*[3]. In this section we investigate a class of non-increasing utility functions in the budget feasibility framework. We will examine a class we call *cut functions*: functions where the value of a subset of agents can be represented as a cut on a graph. This class of functions is a representative of the class of non-increasing utility functions, which, as we now show, has constant factor approximation mechanisms that are budget feasible. The results in this section lead us to conjecture that there are broader classes of non-increasing functions with desirable guarantees in the budget feasibility model.

Definition 15. *A cut function $f : 2^{[n]} \rightarrow \mathbb{R}_{\geq 0}$ is a function for which there exists a graph $G = (\mathcal{N}, E)$ s.t. $f(S) = |C(S)|$, where C is the cut induced by S, i.e. $C(S) = \{(u,v) \in E : u \in S, v \in \mathcal{N} \setminus S\}$.*

[3]In optimization literature this property is referred to as nonmonotnicity. To avoid confusion in discussion of monotonicity of the allocation function, we use the use the term *non-increasing*.

We note that maximizing this function is a variant of the classic Max Cut problem. We denote the degree of a vertex $a_i \in \mathcal{N}$ by d_i (when it will be clear from the context we will use i instead of a_i) and for any $T \subseteq \mathcal{N}$, we use $E(T)$ to denote the set of edges that have at least one vertex in T, i.e. $E(T) = \{(u,v) \in E : u \in T\}$. In our setting each vertex is held by a single strategic agent with a private cost and our objective is to maximize $f(S) = |C(S)|$ under the budget constraint.

Mechanisms for Cut Functions

We first show a randomized mechanism that is budget feasible and obtains a constant factor approximation. We will then discuss its derandomization which also guarantees a constant factor approximation.

A Randomized Mechanism

Consider the following mechanism:

```
A Randomized Mechanism for Cut Functions

1. Set $\mathcal{N}_s = \{a_i \in \mathcal{N} : c_i \leq B/2\}$, $a_s^* \in \text{argmax}_{a_i \in \mathcal{N}_s} d_i$, $S = \{a_s^*\}$ and $i \in \text{argmax}_{i \in \mathcal{N}_s \setminus \{a_s^*\}} \frac{d_i}{c_i}$
2. While $c_i \leq \frac{B}{24} \cdot \left( \frac{|C(S \cup \{a_i\})| - |C(S)|}{C(S \cup \{a_i\})|} \right)$:
   a. Add $i$ to $S$
   b. Set $\bar{\mathcal{N}}$ to be all $j \in \mathcal{N}_s \setminus \{a_s^*\}$ for which $|C(S \cup \{j\})| - |C(S)| \geq \frac{2}{3} d_j$
   c. Set $i \in \text{argmax}_{j \in \bar{\mathcal{N}}} \frac{|C(S \cup \{a_j\})| - |C(S)|}{c_j}$

Output: Choose u.a.r between $S$ and $\text{argmax}_{i \in \mathcal{N} \setminus \mathcal{N}_s} d_i$
```

It is easy to verify that the above mechanism is monotone in the agents' costs and thus truthful. We first prove its approximation guarantee before showing that it is indeed budget feasible. In the following proofs we will use S_i to denote the subset of agents selected by the mechanism after the i^{th} stage.

Lemma 12. *At each stage j we have $|C(S_j)| \geq \frac{1}{3}|E(S_j)|$.*

Proof. We will show by induction on the stage of the mechanism that $|C(S_j)| \geq \frac{1}{3}\sum_{i \in S_j} d_i$ which suffices to prove the lemma since $\sum_{i \in S_j} d_i$ is an upper bound on $|E(S_j)|$.

In the first stage of the mechanism the inequality trivially holds. For a general step j, the vertex a_j that is selected must respect the condition $|C(S_{j-1} \cup \{a_j\})| - |C(S_{j-1})| \geq \frac{2}{3} d_j$. This condition implies that when adding a_j there are at most $\frac{1}{3} d_j$ edges between a_j and vertices in S_{j-1} and thus by adding a_j to S_{j-1} at most $\frac{1}{3} d_j$ edges will be removed from the cut and $\frac{2}{3} d_j$ edges will be added. Thus, together with the inductive hypothesis we have that:

$$|C(S_j)| \geq \Big(|C_{j-1}| - \frac{1}{3}d_j\Big) + \frac{2}{3}d_j \geq \Big(\frac{1}{3}\sum_{i\in S_{j-1}} d_i - \frac{1}{3}d_j\Big) + \frac{2}{3}d_j = \frac{1}{3}\sum_{i\in S_j} d_i$$

□

Lemma 13. *Let S^* be the optimal solution over agents in $\mathcal{N}_s$ with budget $B' = \frac{B}{24}$, then $|C(S)| \geq \frac{|C(S^*)|}{6}$.*

Proof. Partition the set of edges in $C(S^*)$ to the following disjoint subsets of edges: $S_1^* = \{(u,v) \in C(S^*) : u,v \in S\}, S_2^* = \{(u,v) \in C(S^*) : u \in S, v \notin S\}, S_3^* = \{(u,v) \in C(S^*) : u,v \notin S\}$. First, as implied by Lemma 12, we have that $|E(S) \setminus C(S)| \leq 2|C(S)|$ and thus:

$$|C(S_1^*)| \leq |E(S) \setminus C(S)| \leq 2|C(S)| \tag{5.19}$$

In the case of S_2^*, since each vertex has an endpoint in S, it must be that $|C(S_2^*)| \leq |C(S)|$, and thus:

$$|C(S_2^*)| \leq |C(S)| \tag{5.20}$$

In the case where $S_3^* = \emptyset$ the above inequalities suffice to prove our lemma. Otherwise, to bound the ratio between $|C(S_3^*)|$ and $|C(S)|$, assume $S_3^* \neq \emptyset$ and w.l.o.g assume its vertices are labeled s.t. vertex a_i has the greatest ratio between marginal contribution to the cut and cost, given the cut induced by vertices $a_1, \ldots, a_{i-1}$. In such an ordering, for all $i < r = |S_3^*|$ we get:

$$\frac{|C(T_i)| - |C(T_{i-1})|}{c_i} \geq \frac{|C(T_{i+1})| - |C(T_i)|}{c_{i+1}}$$

where T_i is the subset that includes the first i vertices taken according to the ordering and $T_0 = \emptyset$. Let a_k be the first vertex not selected by our mechanism to be in S. In this case we have that:

$$c_k > B' \cdot \Big(\frac{|C(S \cup \{a_k\})| - |C(S)|}{|C(S \cup \{a_k\})|}\Big) \tag{5.21}$$

Since we assume $S_3^* \neq \emptyset$ and $S \cap S_3^* = \emptyset$, all vertices in S_3^* respect the condition of having at least $2/3$ of their edges not connected to vertices in S, and thus such a vertex must exist. Also, since $S_3^* \cap S = \emptyset$, it follows that a_k is either in S_3^*, or that every vertex in S_3^* has smaller marginal contribution ratio to cost than that of a_k. Thus, in either case for any $i \in [r]$ we have that:

$$\frac{c_k}{(|C(S \cup \{a_k\})| - |C(S)|)} \leq \frac{c_i}{d_i} \leq \frac{c_i}{(|C(T_i)| - |C(T_{i-1})|)} \tag{5.22}$$

where the first inequality is due to the fact that the vertices in S_3^* do not have endpoints in S and thus their marginal contribution equals their degree, and the second inequality is due to the decreasing marginal utilities property of the cut function.

Since S_3^* is feasible we have that $\sum_{i=1}^r c_i \leq B'$, which we can write as:

$$\sum_{i=1}^{r}\Big(\frac{c_i}{|C(T_i)|-|C(T_{i-1})|}\Big)\cdot\Big(|C(T_i)|-|C(T_{i-1})|\Big) \leq B'$$

Since $S_3^* = T_r$ we have that $|C(S_3^*)| = \sum_{i=1}^r \Big(|C(T_i)|-|C(T_{i-1})|\Big)$ and therefore together with (5.22) above we have:

$$c_k \cdot \Big(\frac{|C(S_3^*)|}{|C(S\cup\{a_k\})|-|C(S)|}\Big) \leq B'$$

The above inequality, together with (5.21) implies that $|C(S\cup\{a_k\})| > |C(S_3^*)|$. Since S includes the vertex with largest degree it follows that $|C(S)| \geq d_i$, for any $i \in \mathcal{N}_s$, and thus:

$$2|C(S)| \geq |C(S)| + d_k \geq |C(S)\cup\{a_k\}| \geq C(S_3^*) \tag{5.23}$$

To conclude, let $\alpha_i = \frac{|C(S_i^*)|}{|C(S^*)|}$ for $i \in \{1,2,3\}$. Since the sets are disjoint $\alpha_1+\alpha_2+\alpha_3 = 1$. Since there must exist an $\alpha_i \geq 1/3$, from (5.19),(5.20),and (5.23) it follows that $|C(S)| \geq \frac{|C(S^*)|}{6}$. □

Lemma 14. *The mechanism obtains a constant factor approximation ratio in expectation.*

Proof. First, observe that since f is subadditive, for any integer $\alpha > 1$ when the agent a_s^* with largest value is in $OPT(\mathcal{N}_s, B/\alpha)$, then we have that:

$$\alpha \cdot OPT(\mathcal{N}, B/\alpha) + (\alpha-1)f(a_s^*) \geq OPT(\mathcal{N}, B) \tag{5.24}$$

and therefore $(2\alpha-1)OPT(\mathcal{N}, B/\alpha) \geq OPT(\mathcal{N}, B)$.

Since the vertex with largest degree in $\mathcal{N}_s$ is included in S, we have that $47\cdot OPT(c, \mathcal{N}_s, B/24) \geq OPT(c, \mathcal{N}_s, B)$ and thus by Lemma 13 we have that $282|C(S)| \geq OPT(\mathcal{N}_s, B)$. Since we selected the optimal solution in $\mathcal{N}\backslash\mathcal{N}_s$ with probability 1/2 and $OPT(\mathcal{N}, B) \leq OPT(\mathcal{N}_s, B) + OPT(\mathcal{N}\setminus\mathcal{N}_s, B)$ the mechanism is a 564-approximation. □

We will complete the proof of our theorem by showing the mechanism is indeed budget feasible. Unlike the approach taken in the previous where a characterization of payments was shown, we will prove budget feasibility by directly showing a bound on threshold payments.

Lemma 15. *The mechanism is budget feasible.*

Proof. It's easy to see that the threshold payment for a_s^* with largest degree in $\mathcal{N}_s$ is $\frac{B}{2}$, as it is always selected by the mechanism. To bound the threshold payment of the other agents in S by the remaining budget $\frac{B}{2}$, for a given bidding profile $c = (c_1 \dots c_n)$, let a_j be a selected vertex with bid c_j, and let $c'_j > c_j$ be the maximum bid that a_j can declare and remain selected when all other agents declare the same cost, and let $c' = (c_1, \dots, c_{j-1}, c'_j, c_{j+1} \dots, c_n)$. Let S' and S'_i denote the set of selected agents by the mechanism and agents selected at stage i, respectively, when the bid profile is c'.

Let $OPT(c, \mathcal{N}, B)$ denote the value of the optimal solution over the set of agents $\mathcal{N}$ with bid profile c and budget B. First, observe that:

$$|C(S)| \leq OPT(c, \mathcal{N}_s, B') \leq OPT(c, \mathcal{N}_s \setminus \{j\}, B') + d_j \leq 2\Big(OPT(c', \mathcal{N}_s, B')\Big)$$

Together with Lemma 13, the above inequality implies:

$$|C(S)| \leq 2\Big(OPT(c', \mathcal{N}_s, B')\Big) \leq 12|C(S')| \ . \tag{5.25}$$

W.l.o.g. assume that when the bid profile is c agent a_j is selected at stage j. Notice that when running the mechanism with bid profile c', the same first $j-1$ agents are selected as when running the mechanism with the profile c, and we have that:

$$|C(S'_{r-1}) \cup \{a_j\}| - |C(S'_{r-1})| \leq |C(S_{j-1}) \cup \{a_j\}| - |C(S_{j-1})|$$

where r is the stage in which a_j is selected when bidding c'_j. This implies:

$$\frac{c'_j}{(|C(S_j)| - C(S_{j-1})|)} \leq \frac{c'_j}{(|C(S_r \cup \{a_j\})| - C(S_r)|)} \leq \frac{B'}{|C(S')|} \tag{5.26}$$

where the second inequality is due to the fact that every agent $a_i \in S'$ respects the condition $c_i \leq B' \cdot \Big(|C(S'_i)| - |C(S'_{i-1})|\Big)/|C(S')|$. Together, (5.25) and (5.26) imply:

$$c'_j \leq B'\Big(\frac{|C(S_{j-1} \cup \{a_j\})| - |C(S_{j-1})|}{|C(S')|}\Big) \leq 12B'\Big(\frac{|C(S_{j-1} \cup \{a_j\})| - |C(S_{j-1})|}{|C(S)|}\Big)$$

Since c'_j is the maximum bid an agent can declare, it follows that θ_j (the threshold payments for any agent j) are bounded from above by a constant factor of their marginal contribution:

$$\theta_j \leq 12 \cdot B'\Big(\frac{|C(S_j| - |C(S_{j-1})|}{|C(S)|}\Big).$$

Since $|C(S)| = \sum_{j \in S'}(|C(S_j| - |C(S_{j-1})|)$ and $B' = B/24$, the total payments to agents in $S \setminus \{a_s^*\}$ are bounded by $B/2$ which implies budget feasibility. □

Theorem 5.5.1. *For cut functions there is a randomized $O(1)$-approximation mechanism that is universally truthful and budget feasible.*

A Deterministic Mechanism for Cut Functions

The approximation guarantee provided by our mechanism depends on randomizing between the subset S selected in steps (1) and (2) of the above mechanism and the vertex with largest degree in $\mathcal{N} \setminus \mathcal{N}_s$. In order to derandomize the mechanism and provide a bounded approximation guarantee we need to select between the two solutions, based on the value of the cuts they produce. The problem with using a direct comparison between the values of two solutions is the same as in the case of coverage functions from the previous section. We give a concrete example of such a case for cut functions below.

Example: A direct comparison breaks monotonicity. Consider a graph with the disjoint sets of vertices: $\{a_1, a_2, a_3, a_4\}, \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4$, where $|\mathcal{N}_1| = n+2$, $|\mathcal{N}_2| = n$, $|\mathcal{N}_3| = n-1$, $|\mathcal{N}_4| = 3n$, for some integer $n > 100$. For every $i = 1 \dots 4$ each vertex $a \in \mathcal{N}_i$ is only connected to a_i, and a_2 is connected a_3. We will show that there is a cost profile s.t. the mechanism allocates to $\{a_1, a_2, a_3\}$, but as a_2 slightly decreases her cost declaration, a_3 is no longer allocated and since $d_4 = 3n \geq |C(\{a_1, a_2\})| = 2n+2$, a direct comparison will result in a_2 not being selected, and thus breaking monotonicity. For a budget $B = 2$ and costs $c_1 = \epsilon, c_2 = (\frac{n}{n+1})c_3 + \epsilon, c_3 = \frac{n-1}{3n+2} + \epsilon, c_4 = B$, using a small ϵ (say $\epsilon = 2^{-n}$) serves as such an example. In this instance, the mechanism runs the procedure only on $\{a_1, a_2, a_3\}$ and compares its solution with d_4. Under these costs one can verify that the procedure selects $\{a_1, a_2, a_3\}$ which produce a cut with $3n+1 > d_4$. If a_2 reduces her cost to ϵ however, then a_1, a_2 are selected, a_3 is not selected as her marginal contribution now dropped, and since $C(\{a_1, a_2\}) = 2n+2 < d_4$, a_4 will be allocated instead.

Derandomization via Relaxation

To derandomize the mechanism in a manner that preserves monotonicity and provides a constant factor approximation guarantee we use the same approach as in the previous section for coverage functions: rather than a direct comparison between the two solutions, we will compute a linear programming relaxation over $\mathcal{N}_s$, compare between the value returned by this solution and the largest degree in $\mathcal{N} \setminus \mathcal{N}_s$, and select S if and only if the solution returned by the relaxation is greater. As in the coverage case, since the solution returned by the relaxation will be an optimal *fractional* solution, such a scheme guarantees monotonicity: an agent in $\mathcal{N}_s$ that reduces her cost can only increase the value of the optimal fractional solution, thus avoiding the problem discussed in the above example. As long as we can guarantee that the fractional solution returned by the relaxation is a constant factor away from the optimal integral solution over $\mathcal{N}_s$, implementing such a scheme will guarantee a constant factor approximation.

More concretely, the optimization problem can be reformulated as the following integer program:

$$\max \sum_{i<j} z_{ij} \tag{5.27}$$

$$s.t. \; z_{ij} \leq x_i + x_j, \qquad i < j, \tag{5.28}$$

$$z_{ij} \leq 2 - x_i - x_j \qquad i < j, \tag{5.29}$$

$$\sum_{i \in \mathcal{N}} x_i c_i \leq B, \tag{5.30}$$

$$x_i, z_{ij} \in \{0, 1\}, i \in \mathcal{N}, i < j \tag{5.31}$$

where x_i, c_i are variables representing the vertices and their costs, respectively, and z_{ij} represent the edges. As discussed above, we would like to compute a fractional solution in polynomial time that will be a constant factor away from the optimal integral solution of the above program. We will do this by showing that the linear program relaxation has a constant integrality gap. To bound the integrality gap of the LP relaxation of the problem, we assume that $c_i \leq \frac{B}{2}$, for every i.

Theorem 5.5.2. *The LP has a constant integrality gap.*

Proof. To prove the theorem we consider the following randomized rounding algorithm:

```
Randomized Rounding for Cut Functions
1. Add each vertex $a_i$ to the cut $S$ with probability $\frac{x_i}{4}$
2. If $\sum_{i \in S} x_i \cdot c_i > B$ then set $S' = \emptyset$ else $S' = S$

Output: $S'$
```

We will show that $\Pr[(i,j) \in S'] \geq \frac{7}{64} z_{ij}$. Thus, if we let $\mathbf{1}_{ij}$ be the indicator variable that gets a value of 1 when $(i,j) \in S'$ and 0 otherwise, we have that $\mathbb{E}[\mathbf{1}_{ij}] \geq \frac{7}{64} z_{ij}$. By linearity of expectation, $\mathbb{E}[|C(S')|] = \sum_{(i,j) \in E} \mathbb{E}[\mathbf{1}_{ij}] \geq \frac{7}{64} \sum_{(i,j) \in E} z_{ij}$. Therefore there must always be an integral solution that has a value of at least $\frac{7}{64}$ of the value of the optimal fractional solution. We first calculate the probability that $(i,j) \in S$:

$$\begin{aligned}
\Pr[(i,j) \in S] &= \left(1 - \frac{x_i}{4}\right)\frac{x_j}{4} + (1 - \frac{x_j}{4})\frac{x_i}{4} \\
&= \frac{x_i}{4} + \frac{x_j}{4} - 2 \cdot \frac{x_i}{4} \cdot \frac{x_j}{4} \\
&\geq \frac{z_{ij}}{4} - \frac{x_i \cdot x_j}{8} \\
&\geq \frac{z_{ij}}{4} - \frac{(\frac{z_{ij}}{2})^2}{8} \\
&\geq \frac{7}{32} z_{ij}
\end{aligned}$$

where the first equality is by the properties of the randomized rounding, the first inequality is by the LP constraints, and the second inequality by basic analysis and using $z_{ij} \leq x_i + x_j$. The last inequality uses the fact that $z_{ij} \in [0,1]$ and thus $z_{ij} > z_{ij}^2$.

Next we calculate $\Pr[S' = \emptyset | (i,j) \in S]$. If $(i,j) \in S$ this implies that exactly one of the vertices i and j is in S. Assume without loss of generality that $i \in S$. Now $S' = \emptyset$ only if the total budget exceeds B. Observe that $\mathbb{E}[\sum_{i' \in S, i' \neq j, i' \neq i} c_{i'} | i \in S, j \notin S] \leq \frac{B}{2}$ since each i' is selected into S with probability exactly $\frac{x_{i'}}{4}$ and that $\sum_{i \in \mathcal{N}} x_i c_i \leq B$ by the LP constraints. By Markov's inequality:

$$\Pr\Big[\sum_{i' \in S, i' \neq j, i' \neq i} c_{i'} \geq \frac{B}{2} \;\Big|\; i \in S, j \notin S\Big] \leq \frac{1}{2}$$

Taking into account that $c_i \leq \frac{B}{2}$, by our assumption, we can now bound the probability that the budget used by S does not exceed B:

$$\Pr\Big[S' \neq \emptyset \;\Big|\; i \in S, j \notin S\Big] \geq \Pr\Big[\sum_{i' \in S, i' \neq j, i' \neq i} c_{i'} \leq \frac{B}{2} \;\Big|\; i \in S, j \notin S\Big] \geq \frac{1}{2}$$

To conclude, $\Pr[(i,j) \in S] \geq \frac{7z_{ij}}{32}$ and also $\Pr[(i,j) \in S' | (i,j) \in S] \geq \frac{1}{2}$. Thus $\Pr[(i,j) \in S'] \geq \frac{7z_{ij}}{64}$, for every $(i,j) \in E$, as needed by the discussion above. □

We can now formally state the deterministic mechanism. We use $\mathcal{A}$ to denote the allocation rule in steps (1) and (2) of the randomized mechanism, LP to be the optimal fractional solution, and $LP(x)$ to be the value of the LP evaluated on x.

A Deterministic Mechanism for Cut Functions

1. Let $\mathcal{N}_s = \{i :\in \mathcal{N} : c_i \leq B/2\}$, $a^* = \text{argmax}_{i \in \mathcal{N} \setminus \mathcal{N}_s} d_i$
2. Compute $S = \mathcal{A}(\mathcal{N}_s, B)$ and $x^* = LP(\mathcal{N}_s, B)$

Output: if $LP(x^*) \geq f(a^*)$ return S o.w. return $\{a^*\}$

Theorem 5.5.3. *There is a $O(1)$-approximation polynomial time mechanism for cut functions which is truthful and budget feasible.*

Proof. Truthfulness and budget feasibility follow from the arguments in the case of the randomized mechanism. For the approximation guarantee, note that $OPT(c, \mathcal{N}, B) \leq OPT(c, \mathcal{N}_s, B) + OPT(c, \mathcal{N} \setminus \mathcal{N}_s, B)$. Since a^* is the optimal solution in $\mathcal{N} \setminus \mathcal{N}_s$ if its value is larger than $LP(x^*)$ it must be larger than the optimal integral solution as well, and thus choosing a^* guarantees a 2-approximation in this case. Otherwise we have:

$$|C(S)| \geq \frac{OPT(c, \mathcal{N}_s, B)}{282} \geq \frac{7 \cdot L(x^*)}{64 \cdot 282} \geq \frac{|C(\{a\})|}{2579}$$

Therefore in this case, we are guaranteed that $|C(S)| \geq \frac{OPT(c,\mathcal{N},B)}{5158}$. □

5.6 Subadditive Functions

A function $f : 2^{[n]} \to \mathbb{R}$ is subadditive if $f(S \cup T) \leq f(S) + f(T)$. A naïve representation of subadditive functions requires exponential space in n, and thus, as in other such cases we discussed throughout this thesis, we assume our mechanisms are given access to an *oracle* which enables evaluating the function f. Throughout this thesis, whenever a function had exponential representation, we assumed our algorithm has access to a value oracle which receives a subset S and returns $f(S)$. In this section we will use a stronger oracle model, since, as we now show, reasonable approximation ratios cannot be obtained using value oracles.

A Lower Bound in the Value Query Model

Recall that a function $f : 2^{[n]} \to \mathbb{R}$ is called fractionally subadditive if there exists a finite set of additive valuations $\{f_1, \dots, f_t\}$ s.t. $f(S) = \max_{i \in [t]} f_i(S)$. Every submodular function can be represented as a fractionally subadditive function, and all fractionally subadditive functions are subadditive [37]. We will use a straightforward reduction from a lower bound from [39] to show that even in the case of fractionally subadditive functions, it is infeasible to obtain reasonable approximations using value query oracles (regardless of incentive considerations).

Let $\epsilon > 0$, $v(S) = \left(\frac{1+\frac{\epsilon}{2}}{n^{\frac{1}{2}}}\right)|S|$ and for any $T \subseteq [n]$ let $v_T(S) = |S \cap T|$. Observe that both functions are additive. Let $v^*(S) = \max_{T:|T| \leq (1+\epsilon/2)n^{\epsilon}} v_T(S)\}$, and let $f(S) = \max\{v^*(S), v(S)\}$.

Lemma 16 ([39])**.** *There exists a set T, $|T| = \sqrt{n}$, for which distinguishing between the functions f and $\max\{f, v_T\}$ requires using exponentially many value queries in n^{ϵ}.*

If we let $c_1 = c_2, \dots, c_n = 1$ and $B = \sqrt{n}$, for T as in the lemma, we have that $\sum_{i \in T} c_i = B$, and $f'(T) = \sqrt{n}$ while $f(T) = (1 + \epsilon/2)n^{\epsilon}$. Thus, an algorithm that approximates better than the desired approximation ratio must be able to distinguish between these two functions, which requires exponentially many value queries.

A Mechanism for Subadditive Functions

Since value oracles cannot obtain a reasonable approximation, we will consider use stronger oracle model in this section. A *demand oracle* receives a vector of prices $p_1, \dots, p_n$ and returns a subset S s.t. $S \in \text{argmax}_{T \in [n]} f(T) - \sum_{i \in T} p_i$. It is known that demand oracles are stronger than value oracles: a value query can be simulated by a polynomial number of demand queries, though there are examples of demand queries that cannot be simulated with a polynomial number of value queries [10]. Since a value query can be simulated by a polynomial number of demand queries and we're assuming we have access to a demand oracle, we therefore allow our algorithms in this section to use value queries. We assume, without loss of generality, that 1 is the smallest non-zero value of f. We note that in this section we assume that f is increasing.

We start by describing a procedure for finding a bundle of a fixed size t with value close to the bundle of size t with the highest value. For a subadditive function f, let $\mathcal{V} = \{1, 2, 4, \ldots 2^{\lceil \log f(\mathcal{N}) \rceil}\}$. We show that polynomially many demand queries suffice to achieve a 4-approximation[4].

A Procedure for Finding an Approximate Bundle of Size t

```
For each $v$ in $\mathcal{V}$:
  a. Find the bundle $S$ that maximizes the demand when the price per item is $\frac{v}{2t}$
  b. If $f(S) - |S| \cdot t < \frac{v}{2}$ then set $S_v = \emptyset$ and continue to the next $v$
  c. Else, if $|S| > t$, let $S_v$ be some bundle of size $t$such that $S_v \subseteq S$. Else, let $S_v = S$

Output: $(v, S_v)$ for the maximal $v \in \mathcal{V}$ such
    that $S_v$ is not empty
```

The procedure clearly uses a polynomial number of demand queries. Before proving some of its properties and correctness, we make the establish the following claim.

Claim 5.6.1. *Let f be a subadditive function and $S \in argmax_T f(T) - p \cdot |T|$ for some $p \in \mathbb{R}_+$. Then for each $S' \subseteq S$ we have that $f(S') \geq p \cdot |S'|$.*

Proof. From subadditivity we have that:

$$f(S) - p \cdot |S| \leq f(S \setminus S') - p \cdot |S \setminus S'| + f(S') - p \cdot |S'| \tag{5.32}$$

and since S maximizes the demand we have:

$$f(S \setminus S') - p \cdot |S \setminus S'| \leq f(S) - p \cdot |S| \tag{5.33}$$

Together, these inequalities imply that $f(S') - p \cdot |S'| \geq 0$ as required. □

Lemma 17. *Let $S^* \in argmax_{|S|=t} f(S)$. The procedure finds a subset S_v such that $f(S_v) > \frac{f(S^*)}{4}$. Furthermore, v is either the maximal value in $\mathcal{V}$ such that $v \leq f(S^*)$ or v is the minimal value in $\mathcal{V}$ such that $v > f(S^*)$.*

Proof. First, consider an iteration for which $v \leq f(S^*)$. Setting a price $p = v/2t$ for all items, the demand query oracle finds a subset S that maximizes the demand, i.e. $S \in \text{argmax}_T f(T) - p \cdot |T|$. In particular:

[4]In fact, our algorithm can be slightly modified to achieve a $(2 + \epsilon)$-approximation, but the improved approximation algorithm does not suffice for constructing truthful budget feasible mechanisms.

$$f(S) - p \cdot |S| \geq f(S^*) - p \cdot |S^*| = f(S^*) - \frac{v}{2t} \cdot t = f(S^*) - \frac{v}{2} \geq \frac{f(S^*)}{2}$$

Clearly, $f(S) - p \cdot |S| \ \geq \frac{f(S^*)}{2}$ implies $f(S) \geq \frac{f(S^*)}{2}$.

If $|S| \leq t$ then $S_v = S$ and we are already done. If $|S| > t$, by Claim 5.6.1: $f(S_v) \geq p \cdot t = t \cdot \frac{v}{2t} = \frac{v}{2}$. Notice that for the maximal $v \in \mathcal{V}$ where $v \leq f(S^*)$, we have that $v \geq \frac{f(S^*)}{2}$, thus $f(S_v) > \frac{f(S^*)}{4}$.

To finish the proof, consider $v > 2f(S^*)$. Observe that $f(S) - |S| \cdot \frac{v}{2t}$, thus any bundle of size less than t will have a negative profit in Step (b) and the iteration will fail. Thus, assume towards contradiction that the iteration passes Step (b). By our discussion, we have that $|S| > t$. In this case, Claim 5.6.1 gives us that $f(S_v) \geq t \cdot \frac{v}{2t} = \frac{v}{2} > f(S^*)$. A contradiction. □

A Randomized Mechanism

We first use the above procedure to construct a randomized mechanism. In the next subsection we will derandomize this construction to obtain a deterministic mechanism. For this section, let $\alpha = 2 \cdot \lceil \log n \rceil$ and $a^* \in \text{argmax}_{a \in \mathcal{N}} f(a)$.

A Randomized Mechanism for Subadditive Functions

```
For each $t$ in $\mathcal{T} = \{1, 2 \ldots 2^{\lceil \log n \rceil}\}$ in decreasing order:
  a. Let $\mathcal{N}_s$ be the set of items with cost at most $B/(\alpha \cdot t)$ in $\mathcal{N} \setminus \{a^*\}$
  b. Using the procedure, find $(v_t, S_t)$ among items in $\mathcal{N}_s$

Output: Choose u.a.r between $\cup_t S_t$ and agent $a^*$
```

Lemma 18. *The mechanism provides an approximation ratio of* $O(\log^2 n)$.

Proof. Denote by S^* the set of agents participating in the optimal solution. Let $S_1^* = \{a_i \in S^* : c_i > \frac{B}{\alpha}\}$, and $S_2^* = S^* \setminus S_1^*$.

Suppose that $f(S_1^*) \geq \frac{f(S^*)}{2}$. Since the payment for each of the agents in S_1^* is at least $\frac{B}{\alpha}$, $|S_1^*| \leq \alpha$. By subadditivity there must be one bidder $a \in S_1^*$ such that $f(a) \geq \frac{f(S_1^*)}{\alpha}$. In particular we have that $f(\{a^*\}) \geq f(a)$. Thus, if a^* is chosen this gives us an $O(\log n)$ approximation. Since a^* is chosen with probability $\frac{1}{2}$ the expected approximation guarantee is $O(\log n)$ in this case.

Assume now that $f(S_2^*) \geq \frac{f(S^*)}{2}$ (this is the only other case since by subadditivity $f(S_1^*) + f(S_2^*) \geq f(S^*)$). If $f(a^*) \geq \frac{f(S_2^*)}{2}$, then similarly to before we have a constant approximation if a^* is chosen (which happens with probability $\frac{1}{2}$). Otherwise, let $S'_2 = S^*_2 \setminus \{a^*\}$. We have that $f(S'_2) \geq \frac{f(S^*)}{4}$. Now put agents in S'_2 in bins according to their cost s.t. agent $a_i \in S'_2$ is in bin j if and only if $B/(\alpha \cdot 2^{j+1}) \leq c_i < B/(\alpha \cdot 2^j)$ where $j \in \mathcal{T}$. Since there are $O(\log n)$ bins, subadditivity implies that there is a single bin k with value that is at least $O(\log n)$-fraction of $f(S'_2)$. It follows that the optimal solution of size $\alpha \cdot 2^k$ over all items with cost at most $B/(\alpha \cdot 2^k)$ has value at least $O(f(S'_2)/\log n)$. Since one of the iterations of the procedure gives us a set S_k of size 2^k that is a 4-approximation to the solution of size 2^k, and by subadditivity the solution of size 2^k is an $O(\alpha)$-approximation to the solution of size $O(\alpha \cdot 2^k)$, we have that $f(\cup_t S_t) \geq f(S'_2)/4\log^2 n$. Therefore, with probability $\frac{1}{2}$ we have an $O(\log^2 n)$ approximation in this case. □

Lemma 19. *The mechanism is universally truthful.*

Proof. We will show that the mechanism is monotone. Fix the random coin and suppose that agent a^* wins. Notice she will remain selected regardless of her cost, and in particular if she reduces her cost. Assume now that the selected set is $\cup_t S_t$, and consider some agent $a_i \neq a^*$ that wins and reduces her cost from c_i to c'_i. Let $t \in \mathcal{T}$ be the maximal such that $a_i \in S_t$. Notice that a_i will be selected to S_t also if she reduces her cost (since the procedure is oblivious to the actual cost of the agent and takes into account only that the cost is smaller than some threshold). Therefore we have that the mechanism is monotone. □

Lemma 20. *The mechanism is budget feasible.*

Proof. Recall that the payment for an agent is the maximal cost that she can declare and remain allocated[5] (the *threshold cost*). It is not hard to see that if a^* is chosen then her payment is B. For each other agent a_i that is allocated we claim that the threshold cost is at most $\frac{B}{\alpha t_i}$ where t_i is the maximal index such that $a_i \in S_{t_i}$. This implies that the mechanism is budget feasible since for each $t \in \mathcal{T}$, at most t agents may receive a payment of $B/(\alpha \cdot t)$. Thus the total payment in that case is $\sum_{t \in \mathcal{T}} t \cdot B/(\alpha \cdot t) \leq |\mathcal{T}|\frac{B}{\alpha} \leq B$.

We now show that for each other agent a_i that is chosen the threshold cost is at most $\frac{B}{\alpha t_i}$. Suppose agent a_i has cost greater than $\frac{B}{\alpha t_i}$. In this case a_i will not be selected when the size of the bundle considered is t_i or smaller, because her cost is too high. Also a_i will not be selected in iterations in which the bundle size is larger than t_i: in these iterations either the procedure runs on the exact same set $\mathcal{N}_s$ of items (if his cost is still small enough), and for this set of items we know that i is not selected, or a_i has a cost that is too high and thus is not considered for selection at all. □

In conclusion we have the following theorem.

Theorem 5.6.1. *The mechanism is universally truthful, budget feasible, and provides an expected approximation ratio of $O(\log^2 n)$.* □

[5]The mechanism is universally truthful so we fix the random coin and prove the budget feasibility of the mechanism for every outcome of the random coin.

A Deterministic Mechanism

Our next goal is to construct a *deterministic* mechanism with a good approximation ratio. Similar to our previous discussions, selecting the highest-value outcome of the two possible ones breaks incentive compatibility. In this case an agent that reduces her cost might decrease $f(\cup_t S_t)$[6]. Therefore, as in the previous cases we will use a "monotone estimator" for the value of the union that has the property that if the cost of an agent decreases the value of the monotone estimator increases. We make sure that the value of the monotone estimator is "close" to $f(\cup_t S_t)$. Comparing the value of the monotone estimator to the value of single agent with the best value gives us a monotone $O(\log^3 n)$ mechanism. We note that, ignoring computational constraints, one can use the optimal algorithm as a monotone estimator an obtain a deterministic $O(\log^2 n)$ approximation. As before, let $\alpha = 2 \cdot \lceil \log n \rceil$ and $a^* \in \text{argmax}_{a \in mathcalN} f(a)$.

A Deterministic Mechanism For Subbadditive Functions

```
For each $t$ in $\mathcal{T} = \{1, 2 \ldots 2^{\lceil \log n \rceil}\}$ in decreasing order:
  a. Let $\mathcal{N}_s$ be the set of items with cost at most $B/(\alpha \cdot t)$ in $\mathcal{N} \setminus \{a^*\}$
  b. Using the procedure, find $(v, S_t)$, $|S_t| = t$,among items in $\mathcal{N}_s$
  c. Using the procedure, find $(v_t, S)$, $|S| = \alpha t$, among items in $\mathcal{N}_s$

Output: If $\sum_t v_t \geq f(a^*)$ then output $\cup_t S_t$. Else choose only agent $a^*$
```

Lemma 21. *The mechanism provides an approximation ratio of $O(\log^3 n)$.*

Proof. Denote by S^* the set of agents participating in the optimal solution. Let $S_1^* = \{a_i \in S^* : c_i > \frac{B}{\alpha}\}$, and $S_2^* = S^* \setminus S_1^*$. We start by showing two useful facts. The first one is:

$$2 \sum_t v_t \geq f(S_2^*) \tag{5.34}$$

This is because for every t, $2 \cdot v_t \geq f(S_t)$ and from subadditivity. For each t, let S_t^* be the optimal solution of items of size at most $B/(\alpha \cdot t)$. The second inequality is:

$$\sum_t v_t \leq 4 \sum_t f(S_t^*) \leq 4(\log n) \cdot \sum_t f(S_t) \leq 4 \log^2 n \cdot f(\cup_t S_t)$$

[6]This may happen since we do not have exact mechanisms to find the best bundle of size t but rather use an approximation mechanism to do so.

The first inequality follows since v_t is a 4 approximation to S_t^*. The second one follows since f is subadditive and since in S_t^* there are at most a multiplicative factor of $\log n$ more items than in S_t. The third inequality is because t can take at most $\log n$ values.

We now proceed to the proof itself. Suppose that $f(S_1^*) \geq \frac{f(S^*)}{2}$. The payment of each of the agents in S_1^* is at least $\frac{B}{\alpha}$, hence $|S_1^*| \leq \alpha$. Therefore, by subadditivity there must be one bidder $a \in S_1^*$ such that $f(a) \geq \frac{f(S_1^*)}{\alpha}$. In particular we have that $f(a^*) \geq f(a)$. Thus, if a^* is chosen this gives us an $O(\log n)$ approximation. Else, we have that:

$$4\log^2 n \cdot f(\cup_t S_t) \geq \Sigma_t v_t \geq f(a) \geq \frac{f(S_1^*)}{\alpha} \geq \frac{f(S^*)}{2\log n}$$

The first inequality follows from (5.35). This proves that in the case that a^* is not chosen we have an $O(\log^3 n)$ approximation.

Assume now that $f(S_2^*) > \frac{f(S^*)}{2}$ (again, subadditivity implies this is the only other case). If $f(\{a^*\}) \geq \frac{fS_2^*)}{2}$, then similarly to the previous case we have a constant approximation if a^* is chosen. If $\cup_t S_t$ is selected then arguments very similar to (5.35) prove that the approximation ratio is $O(\log^2 n)$ in this case.

Therefore, let $S' = S_2^* \setminus \{a^*\}$ and assume that $f(S') \geq \frac{f(S^*)}{4}$. Using (5.34) and (5.35) together we have that $8\log^2 n \cdot f(\cup_t S_t) \geq f(S_2^*)$, i.e., an $O(\log^2 n)$ approximation. If a^* is selected then, using (5.34), $f(a^*) \geq \Sigma_t v_t \geq \frac{f(S^*)}{4}$, i.e., a constant approximation in this case. □

Lemma 22. *The mechanism is truthful.*

Proof. We will show the mechanism is monotone. Suppose that a^* is allocated. If she reduces her cost she clearly remains allocated, since the expression $\sum_t v_t$ is not affected. Therefore, assume that some agent $a_i \neq a^*$ is allocated and reduces her cost from c_i to c_i'. We will show that the expression $\sum_t v_t$ does not decrease in this case, and monotonicity will follow. Furthermore, we will show that for every t the value of v_t cannot decrease. To see this, fix some t. The only case where the output of the procedure might change is when $c_i > \frac{B}{t}$ but $c_i' \leq \frac{B}{t}$. Notice that the difference is that the procedure considers one more item i. In this case it might happen that the value of the set $f(S_t)$ will go down, but we will show that the value v_t cannot decrease. This follows from the observation that the procedure succeeds for a certain value of $v \in \mathcal{V}$ if it succeeds in step (b): i.e., if there exists a profit maximizing bundle with large enough value. However, if such bundle exists, it will still exist when considering more items. □

The arguments for budget feasibility are almost identical to the arguments in the proof of Lemma 20. Therefore, our theorem follows from all the above lemmas.

Theorem 5.6.2. *The mechanism uses a polynomial number of demand queries, is truthful, individually rational, and budget feasible. Its approximation ratio is $O(\log^3 n)$.* □

Chapter 6

Conclusions

In this thesis we presented impossibility and possibility results of incentive compatible mechanisms. Our investigation was conducted through abstractions of network problems which capture fundamental challenges at the intersection of incentives and computation.

In the first part of the thesis we proved the limitation of incentive compatible mechanisms. We showed that there is a class of problems (combinatorial public projects) that are in APX and have an optimal incentive compatible algorithm, and yet no constant approximation-ratio is achievable by algorithms that are both computationally feasible and incentive compatible. The impossibility result settles the long-standing open question in algorithmic mechanism design regarding the hardness of polynomial time truthfulness. The combinatorial public projects provides a canonical example where there are substantial gaps between truthful and unrestricted algorithms. The approximation and truthfulness boundaries described the second chapter of this part can be the first step towards developing new solution concepts that may provide provable guarantees in cases where incentive compatibility fails.

In the second part of this thesis we showed the possibilities in algorithmic mechanism design. The budget feasibility framework shows that approximation mechanisms can provide strong guarantees and circumvent impossibility results from classical microeconomics. In our main result we saw that for procurement with submodular utility functions, there are budget feasible mechanisms with desirable guarantees. We saw that the only general class where budget feasibility can be obtained is that of subadditive utility functions. We showed poly-logarithmic mechanisms for this class, though whether constant factor budget feasible approximation mechanisms exist for this class is an open question.

Looking forward, the internet now delivers challenges which were in their infancy at the time the work on this dissertation began. Today, petabytes of social and behavioral data are being generated as a result of rapid adoption of online services. The tremendous opportunities and challenges in leveraging this data will require new approaches and will mark a new era in the theory of manipulation-robust algorithms. Hopefully, the results in this thesis will contribute to a good start.

Bibliography

[1] Alexander A. Ageev and Maxim Sviridenko. “Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee”. In: *J. Comb. Optim.* 8.3 (2004), pp. 307–328.

[2] Gagan Aggarwal et al. “Theory research at Google”. In: *SIGACT News* 39.2 (2008), pp. 10–28.

[3] Miklós Ajtai. “The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions (Extended Abstract)”. In: *STOC*. 1998, pp. 10–19.

[4] Aaron Archer and Éva Tardos. “Frugal path mechanisms”. In: *ACM Transactions on Algorithms* 3.1 (2007).

[5] Sanjeev Arora and Carsten Lund. “Hardness of approximations”. In: (1997), pp. 399–446.

[6] Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. “An optimal lower bound for anonymous scheduling mechanisms”. In: *ACM Conference on Electronic Commerce*. 2009, pp. 169–176.

[7] Lawrence M. Ausubel and Paul Milgrom. “The Lovely but Lonely Vickrey Auction”. In: *Combinatorial Auctions, chapter 1*. MIT Press, 2006.

[8] Yossi Azar and Iftah Gamzu. “Truthful Unification Framework for Packing Integer Programs with Choices”. In: *ICALP (1)*. 2008, pp. 833–844.

[9] S. Bikhchandani et al. “Weak Monotonicity characterizes deterministic dominant strategy implementation”. In: *Econometrica* 74(4) (July 2006), pp. 1109–1132.

[10] Liad Blumrosen and Noam Nisan. “Combinatorial Auctions”. In: *Algorithmic Game Theory*. Ed. by Noam Nisan et al. Cambridge University Press, 2007.

[11] Liad Blumrosen and Noam Nisan. “On the computational power of iterative auctions”. In: *ACM Conference on Electronic Commerce*. 2005, pp. 29–43.

[12] Christian Borgs et al. “Multi-unit auctions with budget-constrained bidders”. In: *ACM Conference on Electronic Commerce*. 2005, pp. 44–51.

[13] David Buchfuhrer, Michael Schapira, and Yaron Singer. “Computation and incentives in combinatorial public projects”. In: *ACM Conference on Electronic Commerce*. 2010, pp. 33–42.

[14] Jeremy Bulow, Jonathan Levin, and Paul Milgrom. "Winning Play in Spectrum Auctions". In: *Working Paper* ().

[15] Matthew Cary et al. "Auctions for structured procurement". In: *SODA*. 2008, pp. 304–313.

[16] E. H. Clarke. "Multipart pricing of public goods". In: *Public Choice* 11 (1971), pp. 17–33.

[17] S. Dobzinski, N. Nisan, and M. Schapira. "Approximation Algorithms for Combinatorial Auctions with Complement-free Bidders". In: *STOC*. 2005.

[18] Shahar Dobzinski, Ron Lavi, and Noam Nisan. "Multi-unit Auctions with Budget Limits". In: *FOCS*. 2008, pp. 260–269.

[19] Shahar Dobzinski and Noam Nisan. "Limitations of VCG-Based Mechanisms". In: *STOC*. 2007.

[20] Shahar Dobzinski, Christos H. Papadimitriou, and Yaron Singer. "Mechanisms for complement-free procurement". In: *ACM Conference on Electronic Commerce*. 2011, pp. 273–282.

[21] Shahar Dobzinski and Mukund Sundararajan. "On characterizations of truthful mechanisms for combinatorial auctions and scheduling". In: *ACM Conference on Electronic Commerce*. 2008, pp. 38–47.

[22] Pedro Domingos and Matthew Richardson. "Mining the network value of customers". In: *KDD*. 2001, pp. 57–66.

[23] Edith Elkind, Amit Sahai, and Kenneth Steiglitz. "Frugality in path auctions". In: *SODA*. 2004, pp. 701–709.

[24] Uriel Feige. "A Threshold of ln n for Approximating Set Cover". In: *Journal of the ACM* 45.4 (1998), pp. 634–652.

[25] Uriel Feige. "On Maximizing Welfare when the Utility Functions are Subadditive". In: *STOC*. 2006.

[26] Uriel Feige, Vahab Mirrokni, and Jan Vondrak. "Maximizing non-monotone submodular functions". In: *FOCS*. 2007.

[27] Joan Feigenbaum, Rahul Sami, and Scott Shenker. "Mechanism design for policy routing." In: *PODC*. 2004, pp. 11–20.

[28] Joan Feigenbaum et al. "A BGP-based mechanism for lowest-cost routing." In: *Distributed Computing* 18.1 (2005), pp. 61–72.

[29] Jon Feldman et al. "Budget optimization in search-based advertising auctions". In: *ACM Conference on Electronic Commerce*. 2007, pp. 40–49.

[30] T. Groves. "Incentives in teams". In: *Econometrica* (1973), pp. 617–631.

[31] Kamal Jain and Mohammad Mahdian. "Cost Sharing". In: *Algorithmic Game Theory*. Ed. by Noam Nisan et al. Cambridge University Press, 2007.

[32] Anna R. Karlin, David Kempe, and Tami Tamir. "Beyond VCG: Frugality of Truthful Mechanisms". In: *FOCS*. 2005, pp. 615–626.

[33] David Kempe, Jon M. Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network". In: *KDD*. 2003, pp. 137–146.

[34] Ron Lavi. "Computationally-Efficient Approximation Mechanisms". In: *Algorithmic Game Theory*. Ed. by Noam Nisan et al. Cambridge University Press, 2007.

[35] Ron Lavi, Ahuva Mu'alem, and Noam Nisan. "Towards a Characterization of Truthful Combinatorial Auctions". In: *FOCS*. 2003.

[36] Ron Lavi, Ahuva Mu'alem, and Noam Nisan. *Two Simplified Proofs for Roberts' Theorem*. Submitted. 2004.

[37] Benny Lehmann, Daniel Lehmann, and Noam Nisan. "Combinatorial Auctions With Decreasing Marginal Utilities". In: *ACM conference on electronic commerce*. 2001.

[38] Hagay Levin, Michael Schapira, and Aviv Zohar. "Interdomain routing and games". In: *STOC*. 2008, pp. 57–66.

[39] Vahab S. Mirrokni, Michael Schapira, and Jan Vondrák. "Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions". In: *ACM Conference on Electronic Commerce*. 2008, pp. 70–77.

[40] Elchanan Mossel and Christopher Umans. "On the complexity of approximating the VC dimension." In: *J. Comput. Syst. Sci.* 65.4 (2002), pp. 660–671. URL: `http://dblp.uni-trier.de/db/journals/jcss/jcss65.html#MosselU02`.

[41] Ahuva Mu'alem and Noam Nisan. "Truthful Approximation Mechanisms for Restricted Combinatorial Auctions". In: *Games and Economic Behavior* 64.2 (2008), pp. 612–631.

[42] R. Myerson. "Optimal auction design". In: *Mathematics of Operations Research* 6.1 (1981).

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

[44] Noam Nisan. "Bidding and Allocation in Combinatorial Auctions". In: *ACM Conference on Electronic Commerce*. 2000.

[45] Noam Nisan and Amir Ronen. "Algorithmic Mechanism Design". In: *Games and Economic Behaviour* 35 (2001). A preliminary version appeared in STOC 1999, pp. 166 –196.

[46] Noam Nisan and Amir Ronen. "Computationally Feasible VCG-based Mechanisms". In: *ACM Conference on Electronic Commerce*. 2000.

[47] Christos Papadimitriou. "Algorithms Games and the Internet". In: *the Proceedings of STOC*. 2001.

[48] Christos H. Papadimitriou, Michael Schapira, and Yaron Singer. "On the Hardness of Being Truthful". In: *FOCS*. 2008, pp. 250–259.

[49] Christos H. Papadimitriou and Mihalis Yannakakis. "On limited nondeterminism and the complexity of the V-C dimension". In: *Journal of Computer and System Sciences* (1996).

[50] Kevin Roberts. "The Characterization of Implementable Choice Rules". In: (1979). Ed. by Jean-Jacques Laffont, pp. 321–349.

[51] Michael Saks and Lan Yu. "Weak monotonicity suffices for truthfulness on convex domains". In: *EC*. 2005.

[52] Norbert Sauer. "On the Density of Families of Sets". In: *J. Comb. Theory, Ser. A* 13.1 (1972), pp. 145–147.

[53] M. Schäfer. "Deciding the Vapnik-Cervonenkis dimension is Σ_3^p-complete". In: *Journal of Computer and System Sciences* 58 (1999), pp. 177–182.

[54] Michael Schapira and Yaron Singer. "Inapproximability of Combinatorial Public Projects". In: *WINE*. 2008, pp. 351–361.

[55] Saharon Shelah. "A combinatorial problem; stability and order for models and theories in infinitary languages". In: *Pacific J Math* 41 (1972), pp. 247–261.

[56] Yaron Singer. "Budget Feasible Mechanisms". In: *FOCS*. 2010, pp. 765–774.

[57] Yaron Singer. "How to Win Friends and Influence People, Truthfully: Influence Maximization Mechanisms for Social Networks". In: *WSDM*. 2012.

[58] Kunal Talwar. "The Price of Truth: Frugality in Truthful Mechanisms". In: *STACS*. 2003, pp. 608–619.

[59] W. Vickrey. "Counterspeculation, Auctions and Competitive Sealed Tenders". In: *Journal of Finance* (1961), pp. 8–37.

[60] William Vickrey. "Counterspeculation, Auctions, and Competitive Sealed Tenders". In: *The Journal of Finance* 16.1 (1961), pp. 8–37.

[61] Andrew Chi-Chih Yao. "Some Complexity Questions Related to Distributive Computing". In: *ACM Symposium on Theory of Computing (STOC)*. 1979, pp. 209–213.