The Economics of Internet Protocols

A thesis submitted for the degree of
"Doctor of Philosophy"

by

Michael Schapira

Submitted to the Senate of the Hebrew University

September 2008

This work was carried out under the supervision of

Professor Noam Nisan

To my parents, Daniel and Charlotte Schapira

Acknowledgements

First and foremost, I would like to thank my advisor, Noam Nisan, for his guidance, for introducing me to many of the topics discussed in this thesis, for his openness to new ideas, for being ever-willing to explore new and uncharted territories, for his care and devotion, and, in general, for being the best advisor I could hope for.

I thank Joan Feigenbaum for her remarkable support throughout my graduate studies. Working at Yale with Joan on interdomain routing, and the countless talks we have had on this subject, turned out to have a very significant impact on my research, as reflected in this thesis. Large parts of this thesis are based on research that was done while I was in the US. I thank Scott Shenker and Christos Papadimitriou for hosting me at Berkeley, and for valuable discussions that greatly contributed to this thesis. I thank the members of the computational game theory group at Microsoft Research (Silicon Valley), and, in particular, my host there, Liad Blumrosen. I also thank Rahul Sami for the short, but fruitful, time we spent working together in Ann Arbor.

Last, but certainly not least, I would like to thank my many coauthors. In particular, I thank Aviv Zohar, with whom I have worked a great deal in the past few years on the subjects discussed in this document. I also thank the many others with whom I have worked during my studies: Moshe Babaioff, Giorgos Christodoulou, Shahar Dobzinski, Sharon Goldberg, Jason Hartline, Sam Ieong, Annamaria Kovacs, Hagay Levin, Vahab Mirrokni, Ahuva Mu'alem, Moni Naor, Vijay Ramachandran, Jennifer Rexford, Yaron Singer, Greg Valiant, and Jan Vondrak.

Abstract

The Internet is an arena of complex interplay between computational and economic factors – a computational platform through which various *self-interested, often competing, economic entities interact.* The actual transition of traffic in the Internet is achieved via the operation of few fundamental Internet Protocols (TCP, IP, BGP). These are probably *the most widely-deployed distributed algorithms in existence*, and will most likely continue to be so in the foreseeable future. Internet protocols are required to fulfil both engineering and economic desiderata. It is only natural, then, that *understanding the Internet necessitates the fusion of ideas from computer science, and from the fields of mathematical economics and game-theory.*

During the last decade there has been a surge of interest of the computer science community in such mergers of ideas; motivated by the Internet, computer-science theorists, and networking researchers, have invested a lot of effort into the modeling and the exploration of the intricate connections between the computational and social phenomena in the Internet. The importance of relating ideas from computer-science and from economics, or game-theory, in the networking context, is twofold: On the one hand, economic and game-theoretic concepts and tools help shed light on the operation of Internet protocols, and enable the formulation and achievement of important design goals. On the other hand, networking settings motivate new game-theoretic and economic questions, and networking research offers partial answers to these questions.

Our exploration of the borderline of computer-science theory, networking research, economics and game-theory can be divided into two parts: In the first part of this thesis we focus on the analysis of Internet protocols. In particular, we examine the *Border Gateway Protocol (BGP).* It seems that BGP is the best example to illustrate the inherent connection between computational and economic factors in the Internet. The Internet is comprised of many smaller interconnected networks called *Autonomous Systems (ASes).* ASes vary in size, from large national and multinational networks owned by vast corporations and governments, to small networks servicing a single business. Connectivity between these smaller networks is the raison d'être of the Internet. ASes are not only bound by physical relationships, they are also bound by *business relationships, constituting a huge global market.* BGP is the only protocol, de facto, that handles the task of establishing routes between ASes, known as *interdomain routing.*

We use game-theoretic machinery to address several BGP-related problems, like the central networking problem of guaranteeing BGP's convergence to a "stable" routing outcome, and the important question of ensuring BGP's incentive-compatibility (that has become the paradigmatic problem in the field of distributed algorithmic mechanism design). We close a long-standing open question by providing the first non-trivial *necessary* condition for BGP convergence. We also show that *well-known modifications of BGP, proposed by security researchers, achieve incentive-compatibility without monetary transfers*, thus highlighting interesting connections between security

research and mechanism design. We go on to propose a general game-theoretic framework for the analysis of Internet protocols like BGP. We demonstrate the usefulness of this framework by exhibiting many other (Internet-related and other) applications: adword auctions used by companies like Google and Yahoo!, congestion-control environments inspired by the Transmission Control Protocol (TCP), economic cost-sharing problems, and stable-roommates problems that generalize settings like economic markets and resource allocation problems.

Due to the inherently economic nature of the Internet, an important requirement from Internet protocols (like BGP) is that they be incentive-compatible, i.e., that the different participants be rationally motivated to execute them. In the second part of this thesis we prove several fundamental *impossibility results*, thus clarifying what simply cannot be achieved if we are aiming for incentive-compatibility. *We wish to understand the limitations of combining the computer-science and economics worlds.* Specifically, we wish to understand the obstacles that face us when we attempt to design algorithms (e.g., Internet protocols) that are not only computationally-efficient, but also incentive-compatible. We prove two kinds of impossibility results that reflect two different approaches regarding how to measure the "success" of an algorithm.

Our main impossibility result tackles a question that lies at the very heart of the field of algorithmic mechanism design. *We establish the first significant (non-constant) approximability gap between algorithms that are both incentive-compatible and computationally-efficient, and algorithms that only achieve one of these two desiderata.* This is shown in the context of a novel mechanism design problem which is an abstraction of many common situations, ranging from elections to the design of overlay networks. We provide general techniques for proving such impossibility results. Our result is actually made up of two complementary results – one in the communication-complexity model and one in the computational-complexity model. *Our computational-complexity result is one of the first impossibility results connecting mechanism design to complexity theory*; its novel proof technique involves an application of the Sauer-Shelah Lemma and may be of wider applicability.

Over the past four decades, complexity theory has been successful in classifying optimization problems into various classes, such as P, NP, NP-hard, and many others. Mechanism design is about *incentive-compatible optimization*, in which the inputs are provided by agents who have their own objectives. Our impossibility results can be seen as an attempt to understand the intricate connections between these classic computational complexity classifications and this new regime.

Contents

Chapter 1

Introduction

1.1 The Commercial Internet

Following the birth of the commercial Internet in the early 90s, the Internet has gone through a radical transformation: From a centralized endeavour meant to provide connectivity between cooperative parties, it has metamorphosed into a computational platform through which various *self-interested, often competing, economic entities interact.* The interaction between these entities is very different than that imagined by the Internet's original (academia- and government-employed) designers, ranging from e-commerce to peer-to-peer file sharing. The Internet is comprised of many smaller interconnected networks called *Autonomous Systems (ASes).* ASes vary in size, from large national and multinational networks owned by vast corporations and governments, to small networks servicing a single business. Connectivity between these smaller networks is the raison d'être of the Internet. ASes are not only bound by physical relationships, they are also bound by *business relationships, constituting a huge global market.* The actual transition of traffic in the Internet is achieved via the operation of a few fundamental Internet Protocols (TCP, IP, BGP). These are probably *the most widely-deployed distributed algorithms in existence*, and will most likely continue to be so in the foreseeable future.

The Internet is an arena of *complex interplay between computational and economic factors.* Internet protocols are required to fulfil both engineering and game-theoretic desiderata: From a networking perspective, Internet protocols are required to be simple and easy to execute. Secondly, they must often be computed in a distributed manner, and the computation must be carried out successfully despite the asynchronous nature of the Internet (that can manifest itself in network delays and network malfunctions). Thirdly, due to the vast size of the Internet, the computation should be feasible even if computational nodes have but little information about one another and the topology of the network. Finally, Internet protocols should have good "network complexity", i.e., not require an excessive expenditure in communication, storage space, etc. From an economic perspective, however, utterly different properties of Internet protocols are desirable: First and foremost, we want the computational nodes to be rationally motivated to execute the protocol. That is, we wish these protocols to be *incentive-compatible.* Often, we would also like the outcomes reached by Internet protocols to achieve (or, at least, approximate) some notion of social justice (be it Pareto optimality, maximal social-welfare, max-min fairness, or other such notions defined in the economic literature).

It is only natural that *understanding the Internet necessitates the fusion of ideas from computer* *science, and from the fields of mathematical economics and game-theory.* During the last decade there has been a surge of interest of the computer science community in such mergers of ideas; motivated by the Internet, computer-science theorists, and networking researchers, have invested a lot of effort into the modeling and the exploration of the intricate connections between the computational and social phenomena in the Internet (see, e.g., the seminal works by Nisan and Ronen [76] and by Koutsoupias and Papadimitriou [59]). The importance of relating ideas from computer-science and from economics, or game-theory, in the networking context, is twofold: On the one hand, economic and game-theoretic concepts and tools help shed light on the operation of Internet protocols, and enable the formulation and achievement of important design goals. On the other hand, networking settings motivate new game-theoretic and economic questions, and networking research offers partial answers to these questions. This can lead to the *revisit of classic economic and game-theoretic settings* and the development of new, more refined, concepts and tools.

A striking example for the usefulness of game-theory in computer-science research is the application of the microeconomic theory of mechanism design to the study of computational settings. The objective in mechanism design is designing settings (games) in which it is in the best interest of strategic players to follow a prescribed behaviour (this is sometimes referred to as a reverse engineering approach to game-theory). Algorithmic considerations were first introduced into this setting by Nisan and Ronen [76], who coined the term "*algorithmic mechanism design*" to refer to this new line of research. Algorithmic mechanism design was later extended to distributed computational settings by Feigenbaum, Papadimitriou, and Shenker [36], who founded a subfield called "*distributed algorithmic mechanism design*" [39]. Distributed algorithmic mechanism design, from its very beginning, focused on the analysis of existing Internet protocols, most notably the interdomain routing protocol of today's Internet, namely the *Border Gateway Protocol (BGP).* The paradigmatic problem in this field is identifying realistic settings in which BGP is incentive-compatible [35]. In this thesis, we shall address this important BGP-related problem, as well as several others fundamental interdomain routing problems studied in networking literature. As we shall demonstrate, game-theoretic notions and machinery prove to be very useful in analyzing the dynamics and properties of Internet protocols like BGP.

As previously mentioned, not only is game-theory useful in analyzing networking settings, but the opposite is also true: Networking settings pose unique challenges from a game-theoretic perspective. For instance, while the standard way in mechanism design to incent strategic agents to follow a prescribed behaviour is via monetary payments (either to or by the agents), "real time payments" (like per-packet payments) are often considered undesirable in the Internet. Hence, networking environments often necessitate achieving incentive-compatibility without the use of money. This is a rare phenomenon in mechanism design and is hard to achieve in non-trivial settings. In addition, asynchronous interactions (inherent to the Internet) are little understood in game-theory and economics. These issues, and others, require the further development of game-theoretic ideas. We will show that the existing networking literature, that puts great emphasis on issues like asynchrony, can be of great help in this task.

Our exploration of the borderline of computer-science theory, networking research, economics and game-theory can be divided into two parts: In the first part of our exploration, we analyze Internet protocols to show how the different desiderata posed by the various disciplines can be achieved. In particular, we show how the game-theoretic analysis of central networking environments, like interdomain routing, helps solve crucial networking problems. An important requirement from Internet protocols like BGP is incentive-compatibility. Can this always be achieved? If so, at what cost? In the second part of our exploration we prove several fundamental impossibility results for algorithmic mechanism design, thus clarifying what simply cannot be achieved. In particular, we resolve a long standing open question in algorithmic mechanism design by establishing significant approximability gaps between incentive-compatible, and unrestricted, polynomial-time algorithms. The organization of this thesis reflects this division into two parts.

1.2 Part I: Analyzing Internet Protocols

The first part of the thesis deals with the analysis of Internet protocols. First, we focus on the interdomain routing setting. Then, we propose a general game-theoretic framework for the analysis of settings like interdomain routing.

1.2.1 Interdomain Routing

It seems that the best example to illustrate the previous points is that of the Border Gateway Protocol (BGP). BGP is the only protocol, de facto, that handles the task of establishing routes between ASes, known as interdomain routing. Since not all ASes are directly connected, communication packets often have to traverse several ASes in order to reach their destinations. The packets routes are established via interactions between ASes that enable them to express local preferences over routes, and are affected by the asynchronous nature of the network (message delays, malfunctions, etc.). Since the computation of interdomain routes by BGP is done for every destination AS independently, we can consider a network graph [48] $G = (N, L)$ in which the set of vertices (nodes) N represents the ASes and contains a unique destination node d to whom all other vertices wish to send traffic. Each vertex has strict preferences over all possible simple routes between itself and d.

On a very high level, BGP is an extremely simple (greedy) protocol, in which every AS (node) is requested to repeatedly perform the following actions: Receive update messages from neighbouring vertices announcing their current routes to d. Choose a single outgoing edge to the neighbouring vertex whose route you like the most. Announce your newly chosen route to your neighbours. This process goes on until all ASes do not wish to switch their choices of neighbouring vertices, i.e., until BGP “converges” to a “stable” solution.

Searching for Stability in Interdomain Routing. It was first observed in [98] that even in simple networks the execution of BGP might result in persistent route oscillations. That is, it might be that the repeated myopic update of routes by ASes never converge to a “stable” routing outcome. Networking researchers seek conditions that guaranteed BGP convergence. We tackle this important networking problem, and prove the first non-trivial necessary condition for BGP convergence.

Result: *If there are two stable solutions in the network to which BGP can potentially converge, then there is a timing of update messages that results in persistent route oscillations.*

That is, two stable solutions imply that the network is unstable, in the sense that it could plausibly lead to persistent route oscillations. This result closes a long standing open question, first posed by Griffin and Wilfong [49].

Another question, closely related to the question of whether BGP is guaranteed to converge, is *how long* BGP takes to converge to a stable routing outcome. We present a formal framework for the analysis of BGP's convergence-rate. We consider the class of policies presented by Gao and Rexford [43], which are believed to be a realistic model of policies induced by the commercial relationships between ASes in the Internet. In the celebrated Gao-Rexford setting, every pair of neighbouring ASes can have a customer-provider relationship or a peering relationship. These business relationships induce natural constraints on the ASes' routing policies. We prove a positive result that may account for BGP's good behavior in real-life:

Result: *In the commercial Gao-Rexford setting BGP's convergence rate is roughly twice the length of the longest customer-provider chain (i.e, longest series of ASes in which each AS is the direct customer of the following AS).*

Since the length of the longest customer-provider route in the Internet is estimated to be around 5, this means that only a small constant number of steps is required for convergence.

Security Meets Selfishness: Interdomain Routing and Games. Other than the question of BGP convergence, another natural question is the following: Is it in the best-interest of ASes to adhere to BGP? That is, can an AS better its routing outcome (i.e., be assigned a route that is more highly ranked by it) by unilaterally "deviating" from BGP? In other words: Is BGP incentive-compatible? This problem has gained the status of the paradigmatic problem in distributed algorithmic mechanism design [35].

BGP dynamics, that basically require ASes to make repetitive greedy choices, are very reminiscent of what is known as "best-reply dynamics" in game-theory. Driven by this observation we present the first game-theoretic model of interdomain routing. This model captures many of the intricacies of interdomain routing. In particular, the interaction between the players (ASes) is dynamic and complex – asynchronous, sequential, and based on partial information. Best-reply dynamics in this model indeed capture crucial aspects of BGP. We address the problem of BGP's incentive-compatibility within our novel game-theoretic model.

We show that, unfortunately, BGP is *not* incentive-compatible even in small and realistic networks. In contrast, we show that well-known modifications of BGP, proposed by security researchers, *are* incentive-compatible (without *any* monetary transfers). In fact, we show that these modifications of BGP are even *collusion-proof*, in the sense that even *coalitions* of ASes cannot better the routing outcomes of *all* members in the coalition by deviating from BGP. Our incentive-compatibility results highlight an interesting connection between the two research agendas that deal with lack of compliance in interdomain routing – security research and distributed algorithmic mechanism design.

Result: *Well-studied security enhancements of BGP are incentive-compatible in realistic settings.*

We also consider the standard economic objective of maximizing the social-welfare, that has also been studied in the context of interdomain routing (see [38]). Maximizing the social welfare means finding a routing tree rooted in d, $T = R_1, ..., R_n$, in which node i is assigned route R_i, such that $\Sigma_i v_i(R_i)$ is maximized.

Result: *No computationally-efficient algorithm (and, in particular, BGP) can obtain a good approximation to the optimal social-welfare in the realistic Gao-Rexford setting.*

1.2.2 Best-Reply Mechanisms

The most natural approach for finding a (pure) Nash-equilibrium of a given game is executing "*best reply dynamics*": Start with an arbitrary strategy profile of the players, and in each step let some player switch his strategy to be the best reply to the current strategies of the others. If this process converges, then we have reached a pure Nash equilibrium, and moreover, a "highly justifiable" one. The many qualities of best-reply dynamics (simplicity, distributed nature, low costs in terms of memory, and many more) make it very attractive from an algorithmic perspective. Indeed, many protocols used in practice can essentially be viewed as executions of best-reply dynamics. BGP is but one, though very prominent, of a such a protocol.

The questions discussed above that arise in the interdomain routing setting motivate much broader questions that apply to best-reply dynamics in general. Our exploration of Internet environments like interdomain routing can therefore be regarded as a first step towards the accomplishment of a much more general goal – providing general game-theoretic tools for analyzing different aspects of best-reply dynamics in real-life settings (convergence in asynchronous environments, strategic considerations, etc.).

Best-reply dynamics is the subject of a large body of literature in game-theory and computer-science. Much work has been devoted to questions like "*When is best-reply dynamics guaranteed to converge to a pure Nash equilibrium*?" [85, 68], and "*What is the convergence-rate of best-reply dynamics*?" [27, 26]. We present and address a natural question that has received little, if any, attention: "*When is following best-reply dynamics the best course of action for every player*?".

We propose an abstract family of mechanisms called "best-reply mechanisms" (for which BGP is an example). Best-reply mechanisms turn out to be a generic and extremely useful tool for the design of *computationally-efficient and incentive-compatible* algorithms (protocols) in various important settings. We point out several realistic environments in which best-reply dynamics is incentive-compatible. For all of the examples below, we prove (1) convergence of best-reply dynamics to a pure Nash equilibrium, (2) that best-reply strategies are incentive compatible. In fact, we prove that for many of these cases best-reply dynamics have many other useful properties, like convergence to a Pareto optimal state, guaranteed convergence and incentive-compatibility even in asynchronous settings, and resilience even to manipulations by coalitions of players of any size (*collusion-proofness*). This gives broader context, strengthens, and unifies several known results, and leads to many new ones.

- **Iterative Single-Item Auctions:** This simple example, mainly used to illustrate our framework, deals with a classic economic setting: A single item is sold in a first price auction (with discretized bids).

- **Adword Auctions:** Sponsored search advertising is a major source of revenue for Internet search engines (close to 100 percent of Googles total revenue each year comes from sponsored search advertisements). In adwords auction advertisement slots are sold to competing bidders.

- **Congestion-Control Games:** These games are inspired by the Transmission Control Protocol (TCP).[1]

- **Cost-Sharing Games:** Cost-sharing problems arise in situations in which we wish to distribute the cost of some public service (e.g., building a bridge) between self-interested users

[1] The study of TCP from a game-theory/mechanism design perspective is a long-standing agenda [81, 40, 3].

that will benefit from this service in different extents.

- **Stable-Roommates Games:** This well known example [41] considers how to pair students (say, for the purpose of sharing a dorm room), when each student has his own preference order over possible roommates. Special cases of stable-roommates games include intern assignments to hospitals, and economic markets.

- **Routing Games:** These games are a model of interdomain routing in the Internet (discussed above), where best-reply dynamics model the Border Gateway Protocol (BGP).

1.3 Part II: The Hardness of Achieving Incentive-Compatibility

The second part of the thesis focuses on understanding the limitations of combining the computer-science and economics worlds. Specifically, we wish to understand the obstacles that face us when we attempt to design algorithms that are not only computationally efficient, but also incentive-compatible.

1.3.1 Incentive Compatibility Can Lead to Bad Approximations

We present a novel utilitarian mechanism design problem, called "*combinatorial public project*", which is an abstraction of many real-life settings (e.g., voting, the design of overlay networks). Computationally speaking, this problem is relatively benign; while many of its special cases are NP-hard a constant approximation ratio is achievable in polynomial-time. A classic result in economics states that (for problems like combinatorial public projects) there is a general method for providing incentives for the agents to truthfully reveal their private information, namely the *Vickrey-Clarke-Groves (VCG) mechanisms* [100, 17, 50]. However, VCG requires that we solve *exactly* (many instances of) the optimization problem – an NP-hard task. 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 [75]. In other words, *efficient optimization and incentive compatibility seem to be at loggerheads.* This tension lies at the very heart of algorithmic mechanism design [75, 76, 62, 21].

We establish a huge gap between the quality of the solutions that can be obtained by algorithms for combinatorial public projects that are *both* computationally-efficient *and* incentive-compatible, and by algorithms that satisfy only *one* of these two desiderata. Our inapproximability result settles a long-standing open question in algorithmic mechanism design [40, 86]:

Result: *We exhibit a problem (combinatorial public projects) 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 (does not allow for constant approximations) if we care about both.*

Our result is actually made up of two complementary results – one in the communication-complexity model and one in the computational-complexity model. Technically speaking, each of these results must overcome two main challenges [86]: First, we must provide a combinatorial characterization of truthful algorithms. Then, once we have such a characterization, we can exploit it to prove an inapproximability result.

An interesting way to view our results 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 computational-complexity result essentially states that APX is* not *preserved* (our communication-complexity result proves an analogous statement in the communication model).

1.3.2 Incentive-Compatibility Can Be Costly

Our next impossibility result is based on a different approach to the question of the hardness of incentive-compatibility: Consider the goal of designing truthful algorithms in environments with self-interested players. We seek algorithms that admit the following two preliminary properties: First, *tractability*, which we now interpret in the information-theoretic sense, i.e., a low amount of information needs to be communicated in order to realize the outcome of the algorithm. Second, *incentive compatibility*, i.e., the *existence* of some payment scheme that supports the implementation of the algorithm in equilibrium. We show that tractability and the existence of incentive-compatible payments are insufficient to establish that implementing the algorithm in equilibrium will indeed be simple. This is due to the fact a non-trivial amount of *additional* communication between the different parties may be required in order to compute the equilibrium-supporting prices.

The question of how much overhead one incurs from the computation of incentive-compatible payments was recently introduced by Fadel and Segal [28], who termed this overhead the *communication cost of selfishness.* We show that the amount of communication that is required to compute incentive-compatibility-supporting prices may increase the communication complexity of the algorithm by a factor that is linear in the number of players.

Result: *There are algorithms whose outcome can be computed by communicating x bits, but for which determining both the outcome* and *incentive-compatible prices may require about $n \cdot x$ bits of communication, where n is the number of players.*

We prove that this result holds even for very simple single-parameter domains, where in each possible outcome every player either “wins”or “loses”. Our lower bound is the first evidence that the communication overhead due to the demand for incentive compatibility may be significant. This linear factor (in the number of players) may be substantial in large-scale economic settings, like the Internet and electronic-commerce systems.

1.4 Organization of the Thesis

The thesis is divided into two parts: The first part, that consists of Chapters 2-4 deals with the analysis of Internet protocols. Specifically, Chapter 2 (based on joint work with Rahul Sami and Aviv Zohar [89]) presents the interdomain routing setting and addresses questions regarding BGP convergence. Chapter 3 (based on joint work with Hagay Levin and Aviv Zohar [65]) presents a game-theoretic model of interdomain routing and studies several complexity and incentive-related issues. In Chapter 4 (based on joint work with Noam Nisan and Aviv Zohar [77]) we present the general framework of best-reply mechanisms (that is an abstraction of BGP) and its various applications.

In the second part of the thesis, that consists of Chapter 5-6, we present our impossibility results. In Chapter 5 (based on joint work with Christos Papadimitriou and Yaron Singer [82] and on a result with Ahuva Mu'alem [72]) we present our main impossibility result (regarding the approximability of incentive-compatible algorithms). In Chapter 6 (based on joint work with Moshe Babaioff, Liad Blumrosen and Moni Naor [7]) we present our impossibility result about the communication cost of selfishness.

This thesis deals with a wide variety of issues that necessitate different backgrounds (topics in networking research, mechanism design, communication complexity, computational complexity, approximation algorithms, and more). For ease of exposition, and for fluency, we chose to make the chapters relatively self-contained (even at the cost of presenting the same background information twice), rather than attempt to write a preliminaries chapter that will provide the necessary background for the entire thesis. Hence, the notation and definitions needed for the understanding of each of the chapters are presented within the chapter (for elaborations the reader is sometimes referred to another chapter or to other reading materials). These definitions and notations are consistent throughout the thesis.

Chapter 2

Searching for Stability in Interdomain Routing

2.1 Introduction

In this chapter we address two important questions in networking research regarding the Border Gateway Protocol (BGP): The first question is that of *BGP convergence*, i.e., ensuring that BGP always converge to a "stable" routing outcome. We close a long standing open question by providing the first non-trivial *necessary* condition for such convergence. The second question studied in this chapter is that of *BGP's convergence-rate.* We analyze a formal measure of the convergence time and prove that if the network structure is similar to the current Internet then BGP converges quickly. This may account for BGP's good performance in practice. In the next chapter we discuss other design requirements from BGP, in particular *incentive-compatibility.*

2.1.1 Two Networking Problems

Interdomain routing is the task of establishing routes between the administrative domains, called *Autonomous Systems (ASes)*, that make up the Internet. The *Border Gateway Protocol (BGP)* handles *interdomain routing* in today's Internet. BGP enables ASes to make local decisions based on private *routing policies*, and forms global routes from these local decisions. At a very high level, the execution of BGP requires ASes to continuously make independent *greedy* route selections, based on their local preferences over routes, and announce these choices to all neighbouring ASes. This procedure is very reminiscent to what is known in game-theory as best-reply dynamics. Indeed, we shall elaborate on this way of viewing BGP in the next chapter.

However, as first observed by Varadhan et al. [98], this BGP feature may come at a costly price: The lack of global coordination between local routing policies may result in undesirable routing anomalies, in the form of persistent route oscillations. Hence, a central question regarding BGP is to characterize network instances in which BGP execution will always result in convergence to a stable solution. For this reason, researchers seek conditions that guarantee this BGP *safety* (see the seminal works of Griffin, Shepherd and Wilfong [49, 48], and also [43, 42, 47, 56, 29, 96, 26]).

All known sufficient conditions impose very severe constraints on the routing policies of ASes. In particular, they all imply the uniqueness of a stable routing outcome for the network. Simple examples show that the existence of more than one stable solution in very small specific networks can lead to various routing anomalies like BGP divergence [48, 46]. However, up to this point, researchers have not yet been able to establish this fact for general networks.

In this chapter we address two central questions regarding BGP convergence: The first question is that of BGP safety, and the second question deals with BGP's convergence-rate.

Two stable solutions imply a BGP oscillation: Our first contribution in this chapter is showing that BGP safety *necessitates* the existence of a *unique* stable solution. That is, two stable solution imply that the network is unstable, in the sense that it could plausibly lead to persistent route oscillations. This result thus closes a long standing open question, first posed by Griffin and Wilfong [49].

Our proof requires a different technique than those used in previous works on interdomain routing. In particular, we analyze BGP dynamics using a convenient combinatorial structure, called the *state-transition graph.* The state-transition graph captures a simplified version of BGP dynamics which makes formal analysis easier. We prove that the results we obtain in the simplified model also hold in the standard model of Griffin et al [48]. Our proof techniques suggest that the state-transition graph might be a useful conceptual tool and heuristic for evaluating and designing network configurations. This may assist in the detection of potential route oscillations and their debugging.

Our result provides the first non-trivial *necessary* condition for BGP safety – uniqueness of the stable routing outcome. We note that there is still a big gap between this necessary condition and the known sufficient conditions (like "No Dispute Wheel" [48]). Finding useful characterizations of BGP safety remains an important open question.

Analyzing BGP's running time. Another question, closely related to the question of whether BGP is guaranteed to converge, is *how long* BGP takes to converge to a stable routing outcome. Naturally, we would like BGP to reach a stable routing outcome quickly. The study of BGP's convergence rate was initiated by Labovitz et al. [61]. Answering this question requires a formal definition of how we are to measure this "convergence rate", as the Internet is asynchronous, and particular messages can be delayed or even lost.

We present a formal framework for the analysis of BGP's running time. We define a *BGP phase* to be a period of time in which all ASes are activated, and each AS receives update messages from all its neighbours. We then define BGP's running time for a given instance to be the number of steps it may require to converge.

It is known that BGP can take very long to converge on pathological instances. In this chapter, we seek to analyze the convergence time of BGP under Internet-like settings. In particular, we consider the class of policies presented by Gao and Rexford [43], which are believed to be a realistic model of policies induced by the commercial relationships between ASes in the Internet. In the Gao-Rexford setting, every pair of neighbouring ASes can have a customer-provider relationship or a peering relationship. These business relationships induce natural constraints on the ASes' routing policies.

Our first result is negative: We show that, even with this restricted class of preferences, there are instances such that BGP's convergence rate is linear in the size of the network. Specifically, we show that, in a network with n nodes, it might take BGP n time-steps to converge . We also prove that this lower bound is tight: BGP is always guaranteed to converge in n time-steps.

The linear bound does not bode well, as there are tens of thousands of ASes in today's Internet. Intuitively, however, one would expect BGP to converge at a much quicker rate in practice as ASes' routing policies are local in the sense that they are not influenced by ASes that are far away. We formalize this intuition to prove a positive result that may account for BGP's behavior in real-life: We prove that BGP's running time cannot be more than (roughly) twice the length of the longest customer-provider chain in the network (i.e, longest series of ASes in which each AS is the direct customer of the following AS). Since the length of the longest customer-provider route in the Internet is estimated to be around 5, this means that only a small constant number of steps is required for convergence.

2.1.2 Related Work

Varadhan et al. [98] first observed that BGP policy interaction may lead to persistent route oscillations. As mentioned above, there have been a number of papers that seek to find sufficient conditions to guarantee that BGP converges. In particular, Griffin et al. [48] show that BGP is guaranteed to converge if the set of all ASes routing policies does not contain a particular structure called a Dispute Wheel. Subsequently, Gao and Rexford [43] show that, if each AS's policy conforms to certain restrictions based on its business relationships with other ASes, and the global structure of business relationships consists of a hierarchy with peering, there cannot be a Dispute Wheel, and thus, BGP is guaranteed to converge.

Griffin et al. [48] also proved that the existence of two stable solutions always implies the existence of a Dispute Wheel. Note, however, that this, in itself, is insufficient to imply the existence of a BGP oscillation, because the lack of Dispute Wheels is a sufficient but not a necessary condition for convergence: There are networks which induce a Dispute Wheel, and for which BGP safety is guaranteed.

The study of BGP's convergence rate was initiated by Labovitz et al. [61]. In a synchronous update, Feigenbaum, Sami, and Shenker [38] describe an instance with n nodes in which convergence to the stable solution takes $\Omega(n)$ time steps. However, the hardness in this instance arose because next-hop preferences led to a path of $\Omega(n)$ nodes being selected. In contrast, we show that this convergence time can arise even if no paths with more than 2 hops are ever admitted.

Karloff [57] analyzed the worst-case convergence time of lowest-cost routing policies in a different asynchrony model, in which there can be an arbitrary and adversarial update timing for each node. He shows that, in this model, the number of transitions needed for convergence can even be exponential in the number of nodes. In contrast, our model considers time in the asynchronous system differently: each BGP time step or phase includes an update by each active node (and so we count time according to the slower nodes in the network); this leads to our upper bound of n on the running time.

2.2 The Model

In this section, we describe the formal models that we use to analyze the dynamic behavior of BGP. At the core, this model is similar to the one proposed by Griffin, Shepherd, and Wilfong [49, 48]. However, there are important differences in the way in which the asynchronous execution of BGP is modeled. Our model of the network, policies, and dynamic evolution is described in Section 2.2.1. In Section 2.2.2, we present an alternative way to visualize possible system behavior in this model, by identifying the system evolution with a trajectory in a “state-transition graph”. In Section 2.2.3, we discuss the relation of our model to the standard model of Griffin, Shepherd, and Wilfong [49, 48].

We show that nonconvergence results obtained in our model are valid in the standard model as well.

2.2.1 A Formal Model of BGP dynamics

Network and policies Our model, like the model of Griffin *et al*, abstracts away some implementation-related issues in order to make it easier to analyze BGP convergence. The network is defined by an *AS graph* $G = (N, L)$, where N represents the set of ASes, and L represents physical communication links between ASes. N consists of n *source-nodes* $\{1, \ldots, n\}$ and a unique *destination node* d (in the Internet routes to every destination AS are computed independently). P^i denotes the set of all simple (noncyclic) routes from i to d in G. Each source-node i has a *ranking function* $\leq_i$, that defines a strict order over P^i (that is, i has strict preferences over all routes from i to d). We allow ties between two routes in P^i only if they share the same first link (i, j). The *routing policy* of each node i consists of $\leq_i$ and of i's *import policy* and *export policy.* i's import policy dictates which set of routes $Im(i) \subseteq P^i$ i is willing to send traffic along. We assume that $\emptyset <_i R_i$ for any route $R_i \in Im(i)$ (i prefers any route in $Im(i)$ to not getting a route at all) and that $R_i' <_i \emptyset$ for any route $R_i' \notin Im(i)$ (i will not send traffic at all rather than send traffic along a route not it $Im(i)$). i's export policy dictates which set of routes $Ex(i, j) \subseteq P^i$ i is willing to announce to each neighbour j.

Updates and Activations BGP belongs to an abstract family of routing protocols named *path-vector protocols* [48]. Basically, the routing tree to a given destination is built, hop-by-hop, as knowledge about how to reach that destination propagates through the network. The process is initialized when d announces itself to its neighbours by sending update messages. From this moment forth, every active node establishes a route to d by repeatedly choosing the best route that is announced to it by its neighbours and announcing this route to all of its neighbours via update messages. The network is assumed to be *asynchronous*: ASes can be *activated* in different timings (one can think of the *activation sequence* of ASes as chosen by some adversarial entity [65] – see Chapter 3).

In the model of BGP dynamics put forth by Griffin *et al.* when an activated node chooses a route this is immediately announced via update messages to all of that node's neighbours via update messages (see a game-theoretic interpretation of Griffin *et al.*'s model in Chapter 3). These update messages can be arbitrarily delayed along selective links (or even dropped), as long as no node is indefinitely starved from receiving update messages from one of its neighbours. In our simpler model, there are two kinds of actions that an active node may carry out that potentially change the global routing state:

- A node i may select a route (or change its selected route) from the routes to given destination d that it currently believes to be available.
- A node j may send an update message to a neighbouring node i, informing i of the route that j is currently using to destination d. The update message is assumed to be delivered immediately (without propagation delay), and is immediately reflected in updated beliefs that i has about j's route.

The selection and update actions can occur at arbitrary times. In particular, note that the update messages are not required to be sent at a given time interval, or whenever j's route changes.

For convenience, we assume that only one action takes place at any given time. However, it is important to note that this assumption can be made *without loss of generality* in our model, because the update messages are completely decoupled from the route selections: The effect of a set of concurrent actions (for example, concurrent route selections at a subset S of nodes, followed by concurrent advertisement of updated routes) on the global routing state can be obtained through a particular sequence of actions (an arbitrary sequence of route selections at nodes in S, followed by an arbitrary sequence of update messages from the nodes in S).

We are interested in asking the question of when BGP converges to a stable set of routes. Informally, a stable solution is a global configuration that is not changed by any of the possible actions. Thus, once the system is in a stable solution, there is never any further change (provided the network and policies stay the same). Formally, we define a **stable solution** as an n-tuple of routes in G, $R_1, ..., R_n$, such that:

- **Import constraint:** For every $i \in [n]$ $R_i \in Im(i)$.

- **Export constraint:** For every $i, j \in [n]$ such that $R_i = (i,j)R_j$ (that is, R_i starts with the link $(i,j) \in L$ and then follows R_j) it holds that $R_j \in Ex(j,i)$.

- **Route consistency:** If for two nodes $i \neq j$ it holds that j is on R_i, then it must hold that $R_j \subset R_i$ (that is, R_j is a suffix of R_i).

- **Stability:** If there is a link $(i,j) \in L$, and $R_i \neq (i,j)R_j$, then $(i,j)R_j <_i R_i$.

It is easy to show [48] that a stable solution is always in the form of a tree rooted in d. Further, the import and export policies can be folded into the routing policies, by modifying the preferences so that paths that are filtered out have the lowest possible value. Thus, we do not explicitly consider them in the remainder of this chapter.

At first glance, the dynamic model described in this section appears to be unrealistic in some respects, such as the lack of message delay. Indeed, this high level of abstraction is what leads to the simple and tractable state-transition graph described in the next section, which is critical in proving our main result. However, we show in section 2.2.3 that this additional simplification does not come at the cost of realism: Any nonconvergence results in our model are true in the previous model of Griffin *et. al.* [48], in which ASes can be activated simultaneously and update messages can be arbitrarily delayed along selective links.

2.2.2 The State-Transition Graph

In this subsection, we describe the state transition graph – a tool we use to analyze the convergence of BGP on different instances. We exploit the relative simplicity of our asynchronous update model to summarize all possible dynamic evolutions of BGP as paths in this graph, as defined below.

The *state-transition graph* of an instance of BGP is defined as follows: The graph consists of a finite number of states, each state s is represented by an n-dimensional *forwarding vector* of routing choices $r^s = (r^s_1, ..., r^s_n)$, and a $n * n$ *knowledge matrix* $K^s = \{k^s_{ij}\}_{i,j}$. r^s_i specifies the identity of the node to which node i's traffic is being forwarded, and k^s_{ij} specifies the (loop-free) route that node i *believes* that its neighbouring node j is using. We define $k^s_{ij} = NULL$ when j is not a neighbour of i; any knowledge that i has about non-neighbouring nodes' routes is irrelevant to i's route selection and advertisement decisions. We assume, naturally, that node i knows who it is forwarding traffic to: r^s_i must be the first hop in k^s_{ii}.

A directed edge from a state s to a state s' represents a transition from s to s'. We allow two kinds of atomic actions that lead to transitions:

- *Route transition (route selection actions)*: Informally, a route transition arises when a node i updates its selected route by picking its favorite route from its current knowledge set of routes used by its neighbours.

  Formally, there is an *i-route transition* from state s to state s' iff there is a node i such that: The forwarding vector in s' is identical to the forwarding vector in s with the possible exception of i. Further, $r_i^{s'} = j$ for some node j such that $v_i((i,j)k_{ij}^s \geq v_i((i,j)k_{iu}^s$ for every neighbour u such that $k_{iu}^s \neq NULL \; and \; i \notin k_{iu}^s$. (This ensures that i never picks a route with a loop in it.)

- *Knowledge transition*: Informally, a knowledge transition is an update message sent from a specific node i to a neighbouring node j announcing the route that i believes it is sending traffic along.

  Formally, there is a *ji-knowledge transition* from state s to state s' iff there is a node i and a neighoring node j such that: The forwarding vectors in the two states are identical. The knowledge matrix in s' is identical to the knowledge matrix in s with the exception of i's belief about j, and $k_{ij}^{s'} = k_{jj}^s$. In other words, i learns of the route that j currently believes it is using.

Note that this definition reflects the restricted asynchrony of our dynamic model: We assume that updates can be held up, but when an update is delivered and processed, it reflects the *current* (at the time of delivery) belief of the node that sent the update. We can phrase this restriction equivalently as: Update messages can be delayed in transit, but when they are delivered, a fresh update message from the same sender is delivered immediately (and thus overrides the delayed update.) Thus, the state description does not have to include messages in transit.

Thus, each transition corresponds to one of a set of possible actions $\mathcal{A} = \mathcal{A}^R \cup \mathcal{A}^K$, where $\mathcal{A}^R$ is the set of route selection actions (one for each node) and $\mathcal{A}^K$ is the set of update actions (one for each ordered pair i, j with an edge between i and j). We write $s \xrightarrow{a} s'$ if action a causes a transition from s to s'; we extend the notation to sequences of actions σ, and write $s \xrightarrow{\sigma} s'$ if the eventual (single) state reached by the sequence σ is s'.

In any state, any of the actions in $\mathcal{A}$ may be activated; we include a self-loop $s \xrightarrow{a} s$ if activating action a in state s results in the system staying in the same state.

We shall call the state in which d has not yet announced itself to any of its neighbours the *Zero state.*

Definition 2.1 *The* **Zero state** *is the state s in the state-transition graph in which $r_i^s = \emptyset$ for every node i, and $k_{ij}^s = \emptyset$ for every two nodes i, j.*

Stability and Oscillations in the State-Transition Graph

Definition 2.2 *A* **stable state** *in the state-transition graph is one in which the forwarding vector induces a confluent routing tree T_d to the destination d, the knowledge matrix is such that every node knows its neighbours' routes in T_d, and every node prefers its assigned route over any other available routes.*

In other words, a stable state is one in which the nodes forward traffic along a stable solution, and have complete and accurate knowledge about their neighbours' routes. Observe, that all outgoing transition edges from a stable state lead back to the same stable state. This is because a node will never change its route or acquire new knowledge. For this reason, we shall refer to stable states as *sinks*.

We want to prove the existence of potential BGP oscillations in the state transition graph. In many cases, oscillations occur only for specific timings of asynchronous events. Our aim is to show the existence of an infinite activation sequence that leads to an oscillation. However, while BGP routers have unpredictable delays and timings of individual updates, the activation sequences that arise in practice are not completely arbitrary. In particular, starting at any given point of time, every node *eventually* updates its route selection if its knowledge of routes has changed, and every node *eventually* receives update messages from each neighbour that has changed a route. Moreover, there can only be a finite number of other activations taking place between subsequent route selections (or updates) at a given router. For this reason, we look for oscillations that can arise through a *fair* activation sequence:

Definition 2.3 *An infinite activation sequence σ is said to be* **fair** *if each transition in $\mathcal{A}$ appears infinitely often in σ.*

Definition 2.4 *A* **fair cycle** *in the state-transition graph is a finite cyclic path that does not contain a sink, such that every action in $\mathcal{A}$ is taken at least once in each traversal of the cycle. (Note that the cycle need not be simple, i.e., a state might occur multiple times on a single traversal.)*

To summarize, if an instance of BGP results in a state-transition graph (for a given destination) that has a fair cycle, we will infer that there is a plausible sequence of route selections and updates that will cause BGP to oscillate.

2.2.3 Implications for the model of Griffin *et al*

Our model of the static and dynamic evolution of BGP differs from the model of Griffin *et al* [48] in the simpler model of update messages that does not require the tracking of messages in transit. This simpler model allows us to use the state-transition graph to prove, in section 2.3, that the existence of two stable solutions in an instance of BGP implies the possibility of an indefinite oscillation on a fair sequence of route selections and knowledge updates. It is natural to ask if this result is merely an artefact of the simpler model. In this subsection, we show that this is not the case. We prove that if an instance of BGP permits a fair oscillation in our model, it will also permit a fair oscillation in the model of [48].

We modify the dynamic evaluation model of Griffin *et al* [48] in two ways:

- Messages arrive at their destinations *immediately* (update messages are not delayed).
- In the execution of BGP, it is not necessary that a node inform a neighbouring node of *every* new route it chooses. It must only announce a route it is using every once in a while. Equivalently, we allow for update messages to be dropped.

These modifications are clearly simplifications. At first glance, they may appear to be oversimplifications, but in Subsection 2.2.3, we argue that the results we obtain with the modified model hold in the original model of Griffin *et al* as well. This provides validation for the modified model, and suggests that it could be independently useful in other settings.

Consider the differences in the message model, and the corresponding difference in the notion of a fair activation sequence. In our model, a fair activation sequence is a sequence of knowledge updates and route selections that is asynchronous, but subject to two constraints: (1) Every action is taken infinitely often, and (2) A delayed update message is equated to a dropped update message, in that it is assumed to always be superseded by a fresh update message from the same sender that was not delayed at all. Constraint (1) is equivalent to the fairness constraint of Griffin *et al.* Constraint (2) is different, and at first glance, it seems unnatural. However, any activation sequence in our model that satisfies constraint (2) can be simulated by an activation sequence in the model of Griffin *et al*[1]. Non-convergence results in this restricted model of asynchrony imply the corresponding results *a fortiori* in a more general model of asynchronous activation, such as the one proposed by Griffin et al [48].

2.3 Two Stable Solutions ⇒ BGP Oscillation

In this section we prove our main result, i.e, that if there are two stable solutions then the network is unstable in the sense that persistent route oscillations are possible. We prove the following result:

Theorem 2.1 *If the AS graph G contains two stable solutions, then there is a fair activation sequence under which BGP will oscillate on G.*

That is, two stable solution imply that the network is unstable, in the sense that it could plausibly lead to persistent route oscillations. Therefore, to achieve BGP stability, the network must have a unique stable solution.

The intuition behind our proof is as follows. In the state-transition graph, each stable state will have a corresponding "attractor region": a subset of states (possibly just the stable state itself, or much larger) that, once reached, we can be certain that the system will ultimately converge to the stable state. We can visualize the state-transition graph as a map, with each attractor region a different color – red, blue, etc. However, there will also be some states that do not lie in any one attractor region, because different evolutions from that state could lead to different stable states. We label these states with a distinct color – purple, say – and show that the Zero state must belong in this subset.

The key to the proof is showing that, starting from any purple state, we can find a fair activation sequence that ends at another purple state. We use the properties of route selection and update actions to show that we can swap the order of any two consecutive activations, perhaps repeating one of them, and achieve the same result as the original order. Thus, it is not possible that any given activation a leads to a red state in the original order, but leads to a blue state in the perturbed order. Using this, we show that we can add each activation while staying within the purple region. As the graph is finite, this implies the existence of a fair cycle.

The formal proof of Theorem 2.1 is slightly more involved, because we need to consider multiple cases for the pairs of activations that are perturbed.

Proof: Assume that there are no cycles in the state-transition graph. If so, then, from any state, every fair infinite path must lead to one of the sinks (stable states). Let us assign every sink a

[1] The converse of this statement is not true, but that is not relevant for our proofs, because we only rely on the construction of nonconvergent instances.

unique color that is not purple; for concreteness, we assume that one sink has color red and another has color blue. (There may be other sinks of different colors as well.)

We now extend the coloring to all states in the state-transition graph: Consider a state s. If every fair path from s leads to the same sink, then we shall color s by the color of that sink. If there are two fair activation paths from s that lead to different sinks, we assign s the color purple.

Claim 2.1 *There is at least one purple state.*

Proof: We show that the Zero state is purple. It is sufficient to show that any given stable state – say the red sink – can be reached from the Zero state through a fair activation sequence. We prove this by explicitly constructing such an activation sequence.

Let T be the stable routing tree in the red sink. Starting from Zero, we activate route selections and updates such that each node has its route in T selected. We do this by growing the tree from the destination: First, the destination's neighbours in T are activated to select routes; they will naturally choose the direct link to T, which is the only route available to them. Subsequently, we activate update messages from nodes i already connected to the destination to their children in T, followed by activating the route selection in each child j. This culminates in a state in which all nodes have their routes in T selected. Note that this is not necessarily the red sink state, as nodes may not have complete information about all their neighbours' routes. However, after T has been constructed, we can activate all update and route selection messages in round-robin order, infinitely often. The stability of T guarantees that the routes will never be changed, and thus (in finite time because the graph is finite) we will ultimately reach the red sink state. The finiteness of the initial construction ensures that this activation sequence is fair.

As this construction is possible for each sink state, the Zero state must be purple. ∎

To prove our theorem we need only prove that from every purple node we can reach another purple node by following a path in which all possible actions are activated at least once. Since there is a finite number of purple nodes, if we start from one, construct a fair path to other purple node, and continue this process, eventually a purple node will appear for the second time. Since no purple node is a sink, a path between purple nodes cannot contain a sink, and as all actions were activated in each finite segment constructed, this forms a fair cycle.

Claim 2.2 *From any purple state, there is a finite path containing all actions in $\mathcal{A}$ that leads to another purple state.*

Proof: Consider a purple node p. We want to show that, starting from p, there is a finite activation sequence using every action in $\mathcal{A}$ that leads to a purple state. For contradiction purposes, suppose that such an activation sequence does not exist. Then, there must exist an action a and activation sequence σ (possibly null) such that:

1. $a \notin \sigma$

2. $p \xrightarrow{\sigma} p'$, where p' is purple (possibly $p = p'$).

3. Any finite activation sequence from p' that contains action a results in a state that is not purple.

(If no such a and σ existed, we could add each action in turn to σ while keeping it finite, and still result in a purple node.)

We will now prove that these conditions cannot be satisfied. Consider the action a at state $p_0 = p'$. By the third condition, this must lead to a state q_0 that is not purple; without loss of generality, suppose that q_0 is red. Now, because p_0 is purple, there is some finite path from it leading to a state that is neither purple nor red; without loss of generality, suppose this path is $p_1, p_2, \cdots, p_k$, where $p_1, \cdots, p_{k-1}$ are all purple, and p_k is blue. (The path maybe a single step, in which case p_1 is blue.)

Now, define the corresponding states $q_1, q_2, \cdots, q_k$ as follows: $p_i \xrightarrow{a} q_i$. In other words, q_i is the state reached by action a from p_i. Note that, by condition 3, none of the q_i can be purple. Also, q_0 is red and q_k is blue; thus, there must exist a i such that q_{i-1} is red while q_i is another color (blue without loss of generality).

Let b be the action leading from q_i to q_{i-1}. We consider four cases, and prove a contradiction in each case:

- Case (i): a and b are both route selection actions In this case, a and b must be route selections for distinct nodes. Further, neither a nor b changes the knowledge sets, to the knowledge profile is the same at q_{i-1} and q_i. Now, consider activating the action b from q_{i-i}. As the knowledge states are the same in q_{i-1} and p_{i-i}, we must have $q_{i-1} \xrightarrow{b} q_i$. But this implies that there is a fair activation sequence leading from q_{i-1} to the blue sink, which is impossible because q_{i-1} is red.

- Case (ii): a and b are both knowledge actions In this case, a and b must be updates corresponding to different edges in the network, as they are distinct actions. Hence, a and b affect different components of the global knowledge state K, and thus, the activation sequences ab and ba starting from p_{i-1} lead to the same state. Again, we find that $q_{i-1} \xrightarrow{b} q_i$, leading to a contradiction.

- Case (iii): a is a knowledge action, and b is a route selection action. Let a correspond to an update from some node j to some node i. If b is a route selection at a node other than i, then ab and ba are equivalent, leading to a contradiction as in the first cases. If b is a route selection at i, the situation is a little trickier: The action b might have a different result if it occurred after the action a. Now, consider instead the two activation sequences ab and bab starting at state p_{i-i}. The former activation sequence must lead to a red state, as it passes through q_{i-1}. The latter activation sequence likewise must lead to a blue state, as t passes through q_i. However, the final state should be the same following both activations: In activation sequence bab, whatever route node i chose when action b was first activated at p_{i-1}, the route it chooses in the second activation after a will be the same route that was selected in activation sequence ab. Further, the knowledge sets of all nodes are also the same, as only i's knowledge set has changed in either activation sequence, and i's knowledge set is updated in the same way in both sequences. This leads to a contradiction, as a blue state cannot be a red state.

- Case (iv): a is a route selection action, and b is an knowledge action. This case is symmetric to case (iii); considering the activation sequences ba and aba leads to a contradiction.

Thus, no such a and σ exist; it follows that given any finite activation sequence σ leading from p to a purple node, and given any $a \notin \sigma$, we can extend the activation sequence with a finite number of steps to include a, while still reaching a purple node. Proceeding in this manner, we can construct a finite activation sequence σ that contains all activations in $\mathcal{A}$, such that $p \xrightarrow{\sigma} p*$, where $p*$ is a purple node (perhaps equal to p). ∎ (Concluding proof of theorem 2.1): Repeating this construction, and recognizing that there are finitely many purple nodes, we must eventually return to a state that has been previously visited. Thus, we can construct a finite activation sequence σ such that for some purple node p, $p \xrightarrow{\sigma} p$, and σ contains all actions. By the construction, p is reachable from the Zero state in a finite number of steps. An infinite repetition of this σ after getting to p is a fair activation sequence, and leads to an oscillation, because it never reaches a stable state. ∎

This proof also suggests that the state-transition graph and attractor regions might be a useful conceptual tool for evaluating and designing network configurations. If there is a desired stable state, a routing policy change can be evaluated based on how much it grows the size of the corresponding attractor region. Unfortunately, the state-transition graph is exponential in size, so this is not a useful formal measure for large networks; however, we believe that it could still be a helpful heuristic in designing smaller subnetworks.

2.4 BGP Convergence Time

In this section, we tackle the question of how long BGP takes to converge to the unique stable solution. We begin with the abstract model of BGP as described in section 2.2.1. BGP is an asynchronous protocol, and individual messages may be lost or delayed arbitrarily. As we cannot assume a bound on the actual elapsed time of a single message, any model of convergence "time" needs to define a unit of time measurement that remains meaningful in this asynchronous setting. We propose the following definition:

Definition 2.5 *A* **BGP phase** *is a period of time in which all nodes get at least one update message from each neighbouring node, and all nodes are activated at least once after receiving updates from their neighbours.*

We analyze the number of BGP phases it requires for the network to converge. The rationale for this definition is that, although it is difficult for the analyst to assert numerical bounds on the update frequencies at different nodes, it is reasonable to expect that all nodes are updating at similar timescales. The definition of phases admits asynchrony, thus capturing the realistic possibility that different sequences of update activations can lead to different transient behaviour. At the same time, by tying the unit of measurement to the slowest node's update instead of a fixed time unit (or the fastest update), we avoid pathological worst-case time bounds that are only attained if, for example, one node's update cycle is measured in years instead of seconds or minutes.

How many consecutive phases does it take BGP to converge to a stable solution in the worst case? Routes are propagated through the network one hop at a time, so the best we can hope for is a time proportional to the lenght of the longest route in the stable solution. It is easy to construct instances with n nodes in which there are routes of length $\Omega(n)$. However, these instances are unnatural; currently, internet routes tend to be much shorter than this.

For this reason, we focus on bounding the BGP convergence time on Internet-like. In particular, we consider the class of instances in which nodes have customer/provider or peer relationships, and their preferences satisfy the Gao-Rexford conditions: Studies of the commercial Internet [55] suggest two types of business relationships between pairs of ASes that characterize AS inter-connections: *Customer-provider* or a *peering.* Customer ASes pay their provider ASes for connectivity. Peers, on the other hand, find it mutually advantageous to exchange traffic for free (among their customers). In a seminal paper Gao and Rexford [43] suggest constraints on routing policies that are naturally induced by these business relationships, and prove that these constraints guarantee BGP stability.

- **No customer-provider cycles:** Let G_{CP} be the directed graph with the same set of nodes as G and with a directed edge from every customer to its direct provider. We require that there be no directed cycles in this graph. This requirement has a natural economic justification as it means that no AS is indirectly its own provider.
- **Prefer customers to peers and providers:** A *customer route* of a node i is a route in which i's next-hop node j is i's customer. *Provider* and *peer routes* are defined similarly. We require that nodes always prefer customer routes over peer and provider routes.
- **Provide transit services to customers only:** ASes should export *all* of their routes to their customers, but should only export customer routes to their providers and peers. This is because nodes do not always carry *transit traffic*—traffic that originates and terminates at hosts outside the node. ASes are only obligated (by financial agreements) to carry transit traffic to and from their customers.

The Gao-Rexford constraints are known to be a special case of No Dispute Wheel [42].

Sadly, we show that even in the Gao-Rexford setting, it is possible to come up with examples in which the worst-case convergence time is n BGP phases. Furthermore, this hardness is not solely due to the existence of unnaturally long routes: As the next example illustrates, even if all nodes are only interested in routes of size 2, BGP convergence might require n phases.

Example 2.1 *The graph in Figure 2.1 depicts a network with n nodes, and a destination node d. Node* 1 *prefers to go directly to d. Any other node i prefers the route* $i \rightarrow i-1 \rightarrow d$ *over the direct route* $i \rightarrow d$*. All routes of length greater than* 2 *are less desirable to any node. This set of path preferences is compatible with the Gao-Rexford constraints for the following set of customer/provider relationships:* 1 *is a customer of* 2*,* 2 *is a customer of* 3*, etc.; and, additionally, d is a customer of every other node.*

The figure also depicts an initial routing configuration (using arrows to describe the initial routing choice of each node). In this configuration, node 1 *has no routing choice (perhaps it just came online), all even nodes route directly to the destination and all odd nodes (except* 1*) route through their counter-clockwise neighbour. The reader may verify that this configuration is stable with the exception of node* 1*.*

Now, consider the evolution of the network under the following activation sequence. In each phase, initially all update messages go through, and then all nodes are activated. In the first phase, only node 1 *will change its routing choice and will route to d. In the next phase, only node* 2 *will change it's routing choice and will route through* 1*. Then node* 3 *will change to route through d and so on. The network will eventually converge to the routing outcome in which all odd nodes route directly to d and all even nodes route through their counter-clockwise neighbour.*

Next, we show that this bound is tight for the class of instances that satisfy the Gao-Rexford conditions. In fact, we prove a slightly stronger result: The following proposition shows that our lower bound on BGP's convergence rate is tight on the larger class consisting of all instances in which the "No Dispute Wheel"condition of [48, 42] holds.

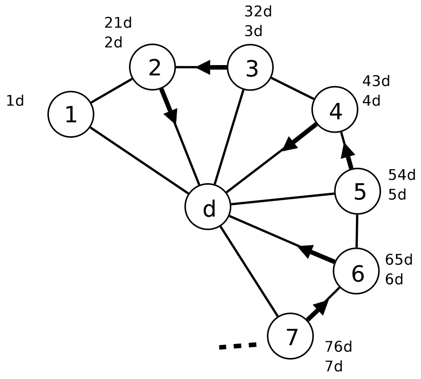


Figure 2.1: A slow-converging network configuration.

Proposition 2.1 *If "No Dispute Wheel" holds then BGP's convergence rate is at most n phases.*

Proof: Let us assume that indeed the "No Dispute Wheel" condition holds in a network graph G with a destination node d. We will show that at every phase, one of the nodes of the graph converges to a route that will not change from that point on. The first node that converges in the first phase is the destination node d, that has the empty path, and announces that path to its neighbours.

We now argue that there must exist a node in the network that is a direct neighbour of the destination d and that its most preferred path is going directly to d. To see that this is indeed the case, pick an arbitrary node v, look at its most preferred path to the destination. This path goes through a neighbour of d right before it reaches d. We shall denote this neighbour by v_1. Now, consider the most preferred path of node v_1, and the closest node to d on that path that we shall denote by v_2. In this manner we define the nodes v_i for $i = 1, 2, 3, \ldots$. At some point, nodes in the sequence we defined must start repeating. If only one node repeats infinitely then this node must have a direct route as its most preferred path, and we are done. Otherwise, the sequence of repeating nodes $v_k, v_{k+1}, \ldots, v_{k+l}$ (for some k, l) forms a dispute wheel: each node prefers to go through the next one in the sequence rather than directly to d. This contradicts our assumption.

Therefore, there exists a node that prefers to go directly to d over any other path. It will choose this path to d on the second phase, send update messages to its neighbours, and will never again change its path (since no path will be better).

We will now continue to follow the convergence process, and observe that at any phase, there must exist a node v that converges to its most preferred route *given the route of the nodes in the system that have already permanently converged.* This node will never again change its path (because unless previous nodes change, it will have no better path, and these previous nodes have also converged). To prove that such a node must exist, we fix the routes of all permanently converged nodes, and pick an arbitrary node v_1 that did not converge. We once again define the sequence of nodes $v_1, v_2, v_3 \ldots$ by defining the node v_{i+1} as the node that is closest to d on the most preferred path of node v_i that did not permanently converge. The set of paths from which we select this most preferred path, is the set of paths that are consistent with the nodes that have already permanently converged. Once again, this sequence of nodes must repeat, and since it cannot contain a dispute wheel, it must have only a single repeating node that is the closest node that did not converge on its own most preferred path. In the next phase, this node's path converges.

We have thus shown that if the AS graph contains no dispute wheels, the convergence time of BGP is bounded by the number of nodes in the entire network graph. ∎

In practice however, we know that the Internet does not converge on a time scale that grows linearly with the size of the network. Our following theorem exhibits a much lower bound on convergence time assuming that the Gao-Rexford [43] economically oriented constraints hold for the network, *and* that the length of customer-provider chains is limited. This is a realistic assumption as it is usually accepted that there are around 3-5 levels in the hierarchy of providers [97]. In fact, a the 3-tier model for the network that is widely accepted as a rough approximation of the structure implies only 3 levels [55]. This in turn implies a very fast convergence time for BGP in the network – even in the worst case.

Theorem 2.2 *In the Gao-Rexford setting, BGP's convergence time is at most* $2\alpha+2$ *phases, where* α *is the length of the longest directed customer-provider chain in the AS graph.*

Proof: We start with some notation and terminology that will assist with the proof. We shall say that a path is *permissible* if every link in the path exists in the network, and will also be reported (i.e., not filtered) by the node, as dictated by the Gao-Rexford constraints.

We say that a node's route is a customer route if the first hop on that route is to one of the node's customers. We define a peer route and a provider route in a similar manner.

Let α denote the length of the longest directed customer-provider chain in the Graph. Because of the Gao-Rexford constraints we are assured that the routing graph has a unique stable state to which the network will converge. We will now prove the following three claims:

1. Any node that has a permissible customer route will converge to a customer route and will do so within $\alpha + 1$ phases.

2. Any node that eventually converges to a peer route does so at most 1 phase after nodes with customer routes have converged.

3. Any node that eventually converges to a provider route will converge within α phases after all nodes with peer and customer routes have converged.

When put together, the three claims imply that the network will converge within $2\cdot\alpha+2$ phases. We now prove them:

1. First, notice that if a node has a permissible customer route then every node on this route forwards its traffic to one of its customers. This is because the Gao-Rexford constraints state that no AS publishes peer or customer routes to any of its providers. Thus, a node that in some permissible path accepts traffic from its provider will not forward it to a peer or provider.

   We will now prove that any node v that has at least one permissible customer route will converge by phase $\alpha_v + 1$ where α_v is the length of v's longest permissible customer route. The proof is by induction. Our base case will be that the convergence time for any node v that has $\alpha_v = 1$ is at most two phases. These nodes are neighbours of the destination. In the first phase, the destination node announces itself and converges to the empty path. In the second phase, any node v for which $\alpha_v = 1$ will select a direct path to the destination. This is because such a node has only a single customer route (which is of length 1) and because it must prefer customer routes to any other route (due to the Gao-Rexford constraints). As this node has no other permissible customer routes, it will never route differently.

In the induction step, we assume that any node v for which $\alpha_v < k$ has converged by phase k. Let u be an arbitrary node for which $\alpha_u = k$. Any node that is on a customer route of u has converged by phase k at the latest, and has updated its neighbours. Therefore, node u's available customer routes will not change from this point on. In phase $k + 1$ node u chooses one of the available customer routes (because the Gao-Rexford constraints dictate that they are preferred over any other route) and never changes its next hop from this point on, as no better path will ever present itself, nor will the current path ever become unavailable (since all nodes on it have also permanently converged). This concludes the proof of the claim.

2. Due to the Gao-Rexford constraints, an AS will not publish its peer routes to other peers or to its providers and therefore any route that begins with a peering connection must proceed as a customer route from the second node and onwards (as the second node on the route would not publish any route other than a customer route to its peer). This implies that the second node on such a route must have a permissible customer route, and therefore has converged by phase $\alpha+1$. On phase $\alpha+2$, any node v with a peer-route that has not already converged will have heard updates from its neighbours about all the peer routes that will ever be available. Since these routes are more preferred than provider routes, the most preferred peer-route will be selected in phase $\alpha + 2$ (or earlier) and will not change from that point on.Note that any peer route that is not available during phase $\alpha + 2$ will never be available, and will never be selected.

3. We denote by β_v the length of the longest customer → provider chain that appears in any permissible path of node v. Surely it for all nodes $\beta_v \leq \alpha$. We now prove the following claim: Any node v that eventually converges to a provider route does so within β_v phases after all nodes with customer and peer routes have converged. We prove this claim by induction on the phase number. Our base case is that one phase after convergence of all customer and peer routes, any node v that has $\beta_v = 1$ converges. Since v's provider on any path is using either a peer route or a customer route, then the next hop of v on any permissible path has already converged, and sent notification messages about its permanent path. v therefore chooses path it finds most appealing and never changes it from this point on.

   In the induction step, we assume that by phase $\alpha + 2 + k$ all nodes v' for which $\beta_{v'} < k$ have already converged. Nodes that have a value of $\beta_v = k$ therefore have their next hop on any permissible path already converged by this time. Thus, they choose the most appealing route, send update messages and never switch paths again.

∎

2.5 Conclusions and Future Work

In this chapter, we studied fundamental questions related to whether and how fast BGP converges to a stable solution. We showed that, in any instance in which the local routing policies can lead to two stable solutions, BGP could potentially lead to persistent oscillations; thus, the existence of a unique stable solution is a necessary condition for safe convergence. We then analyzed the worst-case convergence time of BGP on instances that satisfy the Gao-Rexford conditions. We proved that the convergence time on a graph with n nodes is $\Theta(n)$ in the worst case, but is much smaller in networks with shallow customer-provider hierarchies.

Our results suggest several interesting directions for future work. First, can we close the gap between out necessary condition and known sufficient conditions for safe convergence? Second, can we develop a compositional theory for safe policies: If we put together two subnetworks with unique stable solutions, when does the combination also have a unique stable solution? It would also be valuable to extend the convergence-time analysis to broader classes of preferences, and to characterize the average-case (instead of worst-case) convergence time following a network change. Answers to these questions could provide network operators with new principles to tradeoff the desire for flexible autonomous policies with the need for global routing efficiency.

Chapter 3

Security Meets Selfishness: Interdomain Routing and Games

3.1 Introduction

In this chapter we present a game-theoretic model that captures many of the intricacies of *interdomain routing* in today's Internet. In this model, the strategic agents are source nodes located on a network, who aim to send traffic to a unique destination node. The interaction between the agents is dynamic and complex – asynchronous, sequential, and based on partial information. Best-reply dynamics in this model capture crucial aspects of the interdomain routing protocol de facto, namely the Border Gateway Protocol (BGP).

We study complexity and incentive-related issues in this model. Our main results show that in realistic and well-studied settings, BGP is incentive-compatible. I.e., not only does myopic behaviour of all players *converge* to a "stable" routing outcome, but no player has motivation to unilaterally deviate from BGP. Moreover, we show that even *coalitions* of players of *any* size cannot improve their routing outcomes by collaborating. Unlike the vast majority of works in mechanism design, our results do not require any monetary transfers (to or by the agents).

3.1.1 A Game-Theoretic Approach to Interdomain Routing

As explained in Chapter 2, the Internet is composed of smaller networks called *Autonomous Systems (ASes).* ASes are owned by selfish, often competing, economic entities (Microsoft, AT&T, etc.). The task of establishing routes between ASes is called *interdomain routing.* Since not all ASes are directly connected, packets often have to traverse several ASes. The packets' routes are established via complex interactions between ASes that enable them to express preferences over routes, and are affected by the nature of the network (message delays, malfunctions, etc.). The standard interdomain routing protocol *de facto* is the *Border Gateway Protocol (BGP).*

Routing Games. The first contribution of this chapter is the presentation of a game-theoretic model of interdomain routing that captures many of its intricacies (e.g., the asynchronous nature of the network). Our model can be regarded as a game-theoretic interpretation of the model of Griffin et [49, 48]. In this model, the network is defined by an undirected graph $G = (N, L)$. The set of nodes N represents the ASes, and consists of n *source-nodes* $1, ..., n$ (the players), and a unique *destination-node* d.[1] The set of edges L represents physical communication links between the nodes. Each source node i has a valuation function v_i that expresses a full-order of strict preferences over simple routes from i to d.

The model consists of two games: At the heart of the model is the sequential, asynchronous, and private-information CONVERGENCE GAME, which is meant to model interdomain routing dynamics. Best-reply dynamics in the CONVERGENCE GAME model crucial features of BGP dynamics, in which each AS is instructed to continuously execute the following actions:

- Receive update messages from neighbouring nodes announcing their routes to the destination.
- Choose a single neighbouring node whose route you prefer most (given v_i) to send traffic to.
- Announce your new route to all neighbouring nodes.

We also define a ONE-ROUND GAME, which will function as an analytic tool. The ONE-ROUND GAME can be regarded as the full-information non-sequential game underlying the CONVERGENCE GAME. Pure Nash equilibria in the ONE-ROUND GAME correspond to "stable solutions" in networking literature [49, 48], and are the "sinks" to which best-reply dynamics (BGP) can "converge".

We study several complexity and strategic problems in this model. Most importantly, we address the issue of incentive-compatibility of best-reply dynamics in the CONVERGENCE GAME. We provide realistic settings in which the execution of best-reply dynamics (BGP) is in the best-interest of the players (ASes). We also address the following questions: How hard it is to establish whether a pure Nash exists in the ONE-ROUND GAME (the nonexistence of a pure Nash implies that best-reply dynamics will go on indefinitely)? How hard is it to get good approximations to the optimal social welfare?

Existence of Pure Nash Equilibria. Griffin and Wilfong have shown that determining whether a pure Nash equilibrium in the ONE-ROUND GAME (stable solution) exists is NP-hard [49]. We prove that this result extends to the communication model.

Theorem: *Determining whether a pure Nash equilibrium in the* ONE-ROUND GAME *exists requires exponential communication (in n) between the source-nodes.*

BGP Convergence and Incentives. Networking researchers, and others, invested a lot of effort into identifying sufficient conditions for the existence of a stable solution to which BGP *always* converges (see, e.g., [48, 98, 43, 42, 47, 29, 96, 26]). The most general condition known to guarantee this is "No Dispute Wheel", proposed by Griffin Shepherd and Wilfong [48]. No Dispute Wheel guarantees a *unique* pure Nash in the ONE-ROUND GAME, and convergence of best-reply dynamics to it in the CONVERGENCE GAME. No Dispute Wheel allows nodes to have significantly more expressive and realistic preferences than always preferring shorter routes to longer ones. In particular, a special case of No Dispute Wheel is the celebrated *Gao-Rexford setting* [43, 42] that is said to depict the commercial structure that underlies the Internet [55] (see Section 3.2 for an explanation about No Dispute Wheel and interesting special cases).

Feigenbaum, Papadimitriou, Sami, and Shenker [35] initiated an economic, or mechanism design, approach to interdomain routing. While BGP was designed to guarantee connectivity between trusted and obedient parties, in the age of commercial Internet these are no longer valid assumptions

[1] The reason for this is that in today's Internet, routes to each destination AS are computed independently.

(ASes are owned by different economic entities with very different, and often conflicting, commercial interests). Identifying realistic settings in which BGP is incentive-compatible has become the paradigmatic problem in distributed algorithmic mechanism design (see [39] and references therein), and is the subject of many works [95, 34, 38, 37, 19, 39, 72, 51].

Recently, a step in this direction was taken in [37, 39]. It was shown that if No Dispute Wheel and an additional condition named *Policy Consistency* hold then BGP is incentive-compatible in ex-post Nash (see [95] for an explanation about the ex-post Nash solution concept and its suitability for distributed-computation environments). Informally, policy consistency means that *no* two neighbouring nodes disagree over which of *any* two routes is preferable. This is obviously a very severe restriction that does not necessarily hold in practice. We take a significant step forward by removing it (in particular, we allow the Gao-Rexford commercial setting in which Policy Consistency does not necessarily hold).

Unfortunately, we prove that best-reply dynamics are *not* incentive-compatible if Policy Consistency does not hold. This is true even if No Dispute Wheel holds, and can be shown to hold even in the Gao-Rexford setting.

Theorem: *Best-reply dynamics are not incentive-compatible in ex-post Nash even if the No Dispute Wheel condition holds.*

However, there is still hope for BGP. We define a property called *"Route Verification"*. Route Verification means that a node can verify whether a route announced by a neighbouring node is indeed available to that neighbouring node (and if not simply ignore that route announcement). Unlike Policy Consistency, Route Verification *does not* restrict the preferences of ASes, but is achieved by modifying BGP (e.g., this can be achieved via cryptographic signatures). Achieving Route Verification in the Internet is an important agenda in security research[2]. Security researchers seek ways to implement Route Verification that are not only theoretically sound, but also reasonable to deploy in the Internet (see [15]).

We note that even if announcements of non-available routes are prevented by Route Verification, nodes still have many other forms of manipulation available to them: Pretending to have different preferences ("lying"), conveying inconsistent information (e.g., displaying inconsistent preferences over routes), denying routes from neighbours, and more. Hence, it still needs to be shown that Route Verification guarantees immunity of best-reply dynamics (BGP) to *all* forms of manipulation.

Our main result is the following:

Theorem: *Best-reply dynamics are incentive-compatible in (subgame-perfect) ex-post Nash if No Dispute Wheel and Route Verification hold.*

We stress that this result is achieved *without any monetary transfers* between nodes (as in [39], and unlike most prior works on interdomain routing, and in mechanism design in general).

Our result highlights an interesting connection between the two current research agendas that address the problem of disobedience and lack of trust in interdomain routing – security research and distributed algorithmic mechanism design. One of the implications of this result is that one can achieve incentive-compatibility in realistic settings (e.g., networks for which the Gao-Rexford constraints and Route Verification hold). This should further motivate security research, as it provides a strong strategic justification for modifications of BGP that guarantee Route Verification

[2] "The US government cites BGP security as part of the national strategy for securing the Internet [Department of Homeland Security 2003]" [15]

via cryptographic and other means (e.g., Secure BGP [15]).

In [39] the notion of *collusion-proofness in ex-post Nash* is defined. Informally, collusion-proofness in ex-post Nash means that a *group* of agents cannot collaborate to improve the outcome of any player in the group without strictly harming another player in the group. This means that the group as a whole has no interest to deviate from a strategy profile (at least one member will be harmed by doing so). The previous theorem can actually be strengthened to the following one:

Theorem: *Best-reply dynamics are collusion-proof in (subgame perfect) ex-post Nash if No Dispute Wheel and Route Verification hold.*

In particular, this holds even for the coalition that contains *all* nodes in the network. This implies that, if No Dispute Wheel holds, BGP is actually socially-reasonable in the following sense (see [19]):

Corollary: *If No Dispute Wheel holds, the unique Nash equilibrium in the* One-Round Game *(to which best-reply dynamics always converge) is Pareto optimal.*

Maximizing Social Welfare. Finally, we turn our attention to the objective of maximizing the social-welfare, that has also been studied in the context of interdomain routing (see [38]). Maximizing the social welfare means finding a routing tree rooted in d, $T = R_1, ..., R_n$, in which node i is assigned route R_i, such that $\Sigma_i v_i(R_i)$ is maximized. In [37] it is shown that if No Dispute Wheel and Policy Consistency hold then BGP converges to a stable solution that also maximizes the social welfare. In contrast, we show that the removal of Policy Consistency can be disastrous in terms of welfare maximization. We do so by presenting two complementary bounds, one in the computational complexity model, and one in the communication complexity model.

Theorem: *Obtaining an approximation of $O(n^{\frac{1}{2}-\epsilon})$ to the optimal social welfare is impossible unless $P = NP$. Obtaining an approximation of $O(n^{1-\epsilon})$ to the social welfare is impossible unless $P = ZPP$. This holds for any $\epsilon > 0$ and even if No Dispute Wheel holds.*

Theorem: *Obtaining an approximation of $O(n^{1-\epsilon})$ to the optimal social welfare requires exponential communication (in n). This holds for any $\epsilon > 0$ and even if No Dispute Wheel holds.*

These two bounds actually hold even in the Gao-Rexford setting. These results should be compared with the previously known lower bound of $\Omega(n^{\frac{1}{2}-\epsilon})$ [38] (dependent on $P \neq ZPP$) for the case of general valuation functions. They show that even narrow conditions that ensure existence of pure Nash in the One-Round Game, and convergence of best-reply dynamics in the Convergence Game, might be very far from guaranteeing a good social welfare. A trivial matching upper bound of n exists even for general valuation functions (simply assign the node with the highest value for some route its most desired route).

3.1.2 Organization of the Chapter

In Section 3.2 we present the model and the communication result about pure Nash equilibria. In Section 3.3 we present the results regarding incentives and best-reply dynamics in the Convergence Game. In Section 3.4 we present inapproximability results regarding social-welfare maximization. In Section 3.5 we discuss open questions and directions for future research.

3.2 Routing Games

Here we define the game-theoretic model, and begin exploring the two games it contains.

3.2.1 Two Routing Games

In our model, the network is defined by an undirected graph $G = (N, L)$. N consists of n *source-nodes* $1, ..., n$ (the players), and a unique *destination-node* d. Each source node i has a valuation function v_i that assigns a non-negative value to every *possible* simple route from i to d (i.e., to every simple route in the complete graph over the nodes of G).[3] We make the standard assumption [48] that players have strict preferences: For any node i, and every two routes P, Q from i to d that do not have the same first link, it holds that $v_i(P) \neq v_i(Q)$. The model consists of two routing games:

A benchmark - the One-Round Game:

The One-Round Game is a full-information game in which a strategy of a node i is a choice of an outgoing edge (i's choice of an AS to forward traffic to). The *payoff* of node i for a strategy profile is $v_i(R)$ if the strategies induce a route R from i to d, and 0 otherwise.

The Convergence Game:

The Convergence Game is a multi-round game with an infinite number of rounds. In each round one or more players (nodes) are chosen to participate by a *Scheduler*. The Scheduler models the asynchronous nature of the Internet, and decides which players participate in each round of the game. The *schedule* chosen must allow every player to play in an infinite number of rounds (the Scheduler cannot deny a node from playing indefinitely). In each round of the game, a player i chosen to play can perform the following actions:

- Receive update messages from neighbouring nodes. Each update message announces a simple route from the sending neighbouring node to the destination.
- Choose a *single* outgoing edge $(i, j) \in L$ (representing a choice of a neighbouring node to forward traffic to), or $\emptyset$ (not to forward traffic at all).
- Announce simple routes (from i to d) to i's neighbouring nodes.

The Scheduler decides in which round sent route announcements reach their destinations or if they will be dropped. It can arbitrarily delay update messages, but cannot indefinitely prevent update messages of a node from reaching its neighbour (see [48] for a formal model).

A *strategy* of a player in the Convergence Game specifies his actions in every round in which that player is chosen to participate. *Best-reply dynamics* is the strategy-profile in which every player continuously performs the following actions: Receive the most recent route announcements from *all* neighbours. Choose the neighbour with the *most* preferred simple route to d (according to your v_i). Announce this route to *all* neighbours.

If from some round onwards i's assigned route is constant then i's *payoff* is its value for that route. Otherwise, i's payoff is 0. More formally, the payoff of player i is $v_i(R)$ if R is a simple route

[3] The reason for this is that nodes are not assumed to be familiar with the topology of the network and must be able to express preferences over *all* routes announced by other nodes.

from i to d and from some round onwards, for every link (r, s) on R, r *always* chooses s. Otherwise, i's payoff is 0. (All of our results hold even if when i's route "oscillates" i's payoff is set to be i's value for the highest valued route it is assigned an infinite number of times.).

3.2.2 Stable Solutions and Pure Nash Equilibria

Pure Nash equilibria in the One-Round Game are known in networking literature as stable solutions. It is not hard to verify that each such stable solution forms a tree rooted in d. An important requirement from BGP is that it always converge to such a stable solution. However, this is not guaranteed in general, and definitely will not happen if a pure Nash does not exist. Griffin and Wilfong have shown that determining whether a pure Nash equilibrium in the One-Round Game exists is NP-hard [49]. We strengthen this result by extending it to the communication model. The use of communication complexity for analyzing uncoupled Nash equilibrium procedures was recently presented in [52], following [18]. Our result can be seen as continuing this line of research.

Theorem 3.1 *Determining whether a pure Nash equilibrium in the* One-Round Game *exists requires exponential communication (in n).*

Proof: We shall prove a reduction from the communication Set Disjointeness problem (studied in [4]). In this problem, there are n communication parties. Each party i holds a subset A_i of $\{1, ..., K\}$. The goal is to distinguish between the two following extreme subcases:

- $\bigcap_i A_i \neq \emptyset$ (i.e., all parties have a mutual element)
- For every $i \neq j$ $A_i \cap A_j = \emptyset$ (i.e., no two parties share an element)

It is known [4] that in order to distinguish between these two subcases the parties must exchange $\Omega(K)$ bits (if $K >> n$). We set $K = 2^{\frac{n}{2}}$. The reduction to the problem of determining whether a pure Nash in the One-Round Game exists is as follows: Consider a network with $2n$ source nodes and a unique destination node d. The set of nodes N consists of 2 disjoint subsets: n *sending nodes*, and n *transit nodes*. Each party $i \in [n]$ in the Set Disjointeness problem is associated with a sending node s_i.

The transit nodes are divided into $\frac{n}{2}$ pairs $T_1, ..., T_{\frac{n}{2}}$. Each such pair of nodes T_r contains a specific node we shall call a 0-node and another node we shall call a 1-node. For every $r = 1, ..., \frac{n}{2}-1$, each 0-node in T_r is connected to the 1-node in T_r, and to the 0-node in T_{r+1}. Similarly, each 1-node in T_r is connected to the 0-node in T_r, and to the 1-node in T_{r+1}. Both nodes in $T_{\frac{n}{2}}$ are connected to each other and directly to d. We divide the sending nodes into groups of size 3, $S_1, ..., S_{\frac{n}{3}}$. For every S_r, the 3 nodes in S_r are connected to each other, to the 0-node in T_1, and to d. See a description of the network in Figure 3.1. We shall refer to the 0-node in T_1 (which is the access point of all source-nodes to the transit nodes) as c.

We must now define the valuation functions of the different nodes. Let us start with the transit nodes: A 0-node in T_r u has a value of $\frac{1}{2}$ for any route to d in which the next-hop node after u is the 1-node in T_r, and a value of $\frac{1}{4}$ for routes in which the next-hop node after u is the 0-node in T_{r+1} (and very low values for all other routes to d, without ties). Similarly, a 1-node in T_r u has a value of $\frac{1}{2}$ for any route to d in which the next-hop node after u is the 0-node in T_r, and a value of $\frac{1}{4}$ for routes in which the next-hop node after u is the 1-node in T_{r+1} (and very low values for all

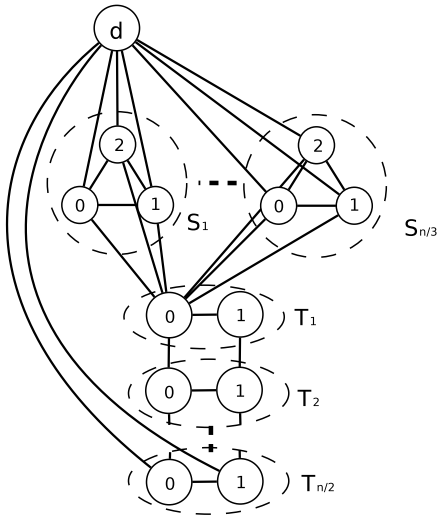


Figure 3.1: Network constructions used in the proof of Theorem 3.1

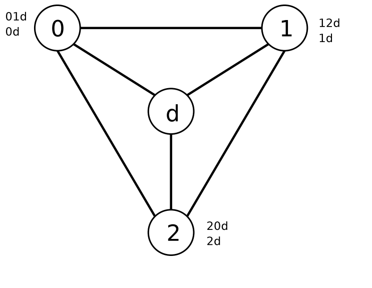


Figure 3.2: Bad Gadget

(a) Network structure for the proof (b) Bad Gadget

other routes to d, without ties). Both nodes in $T_{\frac{n}{2}}$ prefer going through each other (a value of $\frac{1}{2}$) than directly to d (a value of $\frac{1}{4}$).

Fix a specific triplet of nodes S_r and let $0, 1, 2$ be those nodes. Each node $i = 0, 1, 2$ will assign a value of $\frac{1}{2}$ to the route $(i, i + 1\ (mod\ 3), d)$, a value of $\frac{1}{4}$ to the direct route to d, and very low values (without ties) to all other simple routes to d that only include nodes in S_r. This construction of S_r is known as Bad Gadget [49], and appears in Figure 3.2. Bad Gadget is an example of a small network in which no pure Nash equilibrium exists.

There are $2^{\frac{n}{2}}$ possible routes that go through the transit nodes and correspond to strings in $\{0, 1\}^n$: Consider a string of bits $b = (b_1, ..., b_{\frac{n}{2}}) \in \{0, 1\}^{\frac{n}{2}}$. b corresponds to the route in which the b_i-node in T_i forwards traffic to the other node in T_i. Fix an arbitrary order over these routes $R_1, ..., R_{2^{\frac{n}{2}}}$. Now, let each source node s_i assigns a value of 1 (no ties) to the route $((s_i, c)R_a$ iff $a \in A_i$. s_i assigns 0 to all unmentioned routes (observe that since all these routes go through the same next-hop node c breaking ties is not a problem).

The key point is that once the route of c is fixed, all source-nodes have no other choice of route through the transit nodes. Hence, either all source nodes agree on a route that goes through the transit nodes, or some choose not to route through the transit nodes at all. The reader can verify that if there is some $a \in \bigcap_i A_i$ then assigning every node i the route $(s_i, c)R_a$ is a pure Nash equilibrium (and, in fact, a unique pure Nash equilibrium). If, on the other hand, for every $i \neq j$ $A_i \cap A_j = \emptyset$, then there is no pure Nash (because of the Bad Gadget construction for every triple S_r).

Hence, we have a network with $O(n)$ nodes, in which determining whether a pure Nash equilibrium exists is equivalent to solving the Set Disjointeness problem with n players (each player i simulates node s_i), each holding a subset of $1, \ldots, 2^{\frac{n}{2}}$. It therefore requires at least $\Omega(2^{\frac{n}{2}})$ bits of communication. This concludes the proof of the theorem. ∎

We note that the construction we use in our proof is unlikely to appear in real-life networks; in particular, we require nodes to have unreasonable preferences, according to which very long routes (linear in n, the number of all ASes in the entire Internet) are sometimes very desirable. However, the machinery we use in the proof is sufficient to show the following:

Theorem 3.2 *Let the valuation functions of the nodes be such that nodes always strictly prefer routes of length at most x to routes of length greater than x. Determining whether a pure Nash equilibrium in the* One-Round Game *exists requires $\Omega(2^x)$ bits.*

The theorem is proven by using the same construction as in the proof of Theorem 3.1, only with a transit node hierarchy of depth x instead of $\frac{n}{2}$. This result has interesting implications for BGP convergence, as it shows a lower bound on the worst-case amount of communication that BGP (or any other protocol) requires in order to reach a stable solution, as a function of the nodes' valuations.

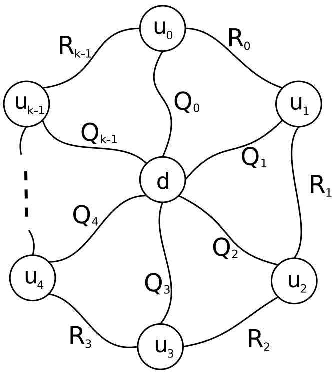
3.2.3 BGP Convergence and Best-Reply Dynamics

No Dispute Wheel

No Dispute Wheel is the broadest condition known, to date, to guarantee BGP convergence to a stable solution. In our terms, this translates to convergence of best-reply dynamics in the Convergence Game. A Dispute Wheel, defined by Griffin et al. [48], is an abstract mathematical structure that can be induced by the network topology and the valuation functions. Formally, a Dispute Wheel is defined as the 3-tuple $(\mathcal{U}, \mathcal{R}, \mathcal{Q})$ where $\mathcal{U} = (u_0, u_1, \ldots, u_{k-1})$ is a sequence of k nodes in the network and $\mathcal{R} = (R_0, R_1, \ldots, R_{k-1})$, $\mathcal{Q} = (Q_0, Q_1, \ldots, Q_{k-1})$ are sequences of routes that exist in G (indices for these nodes and routes should be considered modulo k). We shall call $u_0, \ldots u_{k-1}$ the *pivot-nodes*. It must hold that:

- Each route Q_i starts at u_i and ends at the destination node d.
- Each route R_i starts at node u_i and ends at node u_{i+1}.
- $v_i(Q_i) \leq v_i(R_i Q_{i+1})$ (where $R_i Q_{i+1}$ is the concatenation of the routes R_i and Q_{i+1})

The term "Dispute Wheel" is due to the resemblance of its form to that of a wheel. Figure 3.3 depicts such a structure. It is known that No Dispute Wheel guarantees that there is a unique stable solution [48].



Each node u_i would rather route clockwise through node u_{i+1} than through the path Q_i

Figure 3.3: A Dispute Wheel

No Dispute Wheel is known to hold for several interesting special cases. One special case is *Metric-Based Routing* (defined in [37]) which is a generalization of Shortest-Path Routing (in which the length of every link is 1).

In practice, there is no objective metric according to which all route choices are made. ASes' preferences are influenced by many economic, and other, considerations, like the business relationships between them. The Gao-Rexford setting is said to accurately depict the underlying commercial structure of today's Internet [55], and is a special case of No Dispute Wheel [42]. The Gao-Rexford setting was presented in Chapter 2. For completeness we present it again here.

The Gao-Rexford Framework

Studies of the commercial Internet [55] suggest two types of business relationships that characterize AS inter-connections: Pairs of neighbouring ASes have either a *customer-provider* or a *peering* relationship. Customer ASes pay their provider ASes for connectivity – access to Internet destinations through the provider's links and advertisement of customer destinations to the rest of the Internet. Peers are ASes that find it mutually advantageous to exchange traffic for free among their respective customers, *e.g.*, to shortcut routes through providers. An AS can be in many different relationships simultaneously: It can be a customer of one or more ASes, a provider to others, and a peer to yet other ASes. These agreements are assumed to be longer-term contracts that are formed because of various factors, e.g., the traffic pattern between two nodes.

In a seminal paper Gao and Rexford [43] suggest constraints on routing policies that are naturally induced by the business relationships between ASes.

1. **No customer-provider cycles:** Let G_{CP} be the directed graph with the same set of nodes as G and with a directed edge from every customer to its direct provider. We require that there be no directed cycles in this graph. This requirement has a natural economic justification as it means that no AS is indirectly its own provider.

2. **Prefer customers to peers and peers to providers:** A *customer route* is a route in which the *next-hop* AS (the first AS to which packets are forwarded on that route) is a customer. *Provider* and *peer routes* are defined similarly. We require that nodes always prefer customer

routes over peer routes, which are in turn preferred to provider routes. This constraint is on the valuation functions of the nodes – it demands every node assign customer routes higher values than peer routes, which should be valued higher than provider routes.

3. **Provide transit services only to customers:** Nodes do not always carry *transit traffic*—traffic that originates and terminates at hosts outside the node. ASes are obligated (by financial agreements) to carry transit traffic to and from their customers. However, ASes do not carry transit traffic between their providers and peers. Therefore, ASes should share only customer routes with their providers and peers but should share *all* of their routes with their customers. This constraint is on the filtering policy of the nodes – it requires that nodes only export peer and provider routes to their customers (customer routes are exported to all neighbouring nodes).

3.3 Best-Reply Dynamics and Incentives

In this section we discuss several results regarding the incentive-compatibility of best-reply dynamics (BGP) in the Convergence Game.

3.3.1 BGP Is Not Incentive-Compatible

We first prove the following theorem:

Theorem 3.3 *Best-reply dynamics are not incentive-compatible in ex-post Nash even if the No Dispute Wheel condition holds.*

Proof: Consider the network in Figure 3.4. There are 3 source nodes, $1, 2, m$ and a destination node d. Each nodes' two most preferred routes are listed next to it (where the higher route is more preferred than the lower route). It is easy to show that the valuation functions and topology of the network do not induce a Dispute Wheel (because route $2md$ does not actually exist). Hence, there is a unique pure Nash in the One-Round Game, and best-reply dynamics always converge to this Nash in the Convergence Game. The unique pure Nash is assigning $12d, 2d, m12d$ to $1, 2, m$, respectively. However, if m announces to 2 repeatedly that its route is md then it is easy to see that best-reply dynamics will always assign 1 the direct route to d, $1d$, thus enabling m to get its most preferred route $m1d$. Hence, deviating from best-reply dynamics (violating BGP) is beneficial.

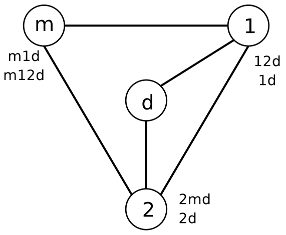


Figure 3.4: Best-Reply Dynamics Are Not Incentive-Compatible

We note that it is possible to define business relationships between the nodes in this example that are consistent with the Gao-Rexford constraints; for instance, assume that m is a customer of $1, 2$, who are both customers of d, and that 1 is a customer of 2. Observe that this is consistent with the first two Gao-Rexford constraints – it does not create customer-provider cycles, and the preferences of all nodes are such that customer routes are always preferred to provider routes (there are no peer routes). Even if we assume that $1, 2$ (and d) execute BGP and uphold the third Gao-Rexford constraint (they only announce peer- and provider-routes to their customers), m can still gain by lying to 2, as explained above. ∎

3.3.2 BGP With Route Verification Is Incentive-Compatible

As we have said before, the fact that BGP, as is, is not incentive compatible, can be rectified if Route Verification holds.

Theorem 3.4 *If No Dispute Wheel and Route Verification hold, then best-reply dynamics are incentive-compatible in (subgame perfect) ex-post Nash*

Proof: Consider a network graph $G = (N, L)$ for which No Dispute Wheel holds. There is a unique stable solution T to which all best-reply dynamics are bound to converge in the Convergence Game. We denote the route of every source node r in T by T_r.

Assume, by contradiction, that some manipulating node r_m manages to reach a different outcome M by unilaterally deviating from best-reply dynamics (not executing BGP), and gains by doing so. We shall show that this implies the existence of a Dispute Wheel. The proof shall proceed in steps, pointing out a sequence of routes in the graph that will eventually form a Dispute Wheel.

We define the route M_r to be the route node r *believes* it is assigned in M. That is, it could be that the manipulator tricked nodes that send traffic through it in M to believe that their traffic is forwarded along a route not used in practice. We note that it could be the case that node r_m intentionally causes a protocol divergence that does not affect it, in order to improve its routing outcome (the choices of other nodes cause remote persistent route oscillations). If this is the case then the route M_r of a node r that is affected by this divergence, will simply be r's most preferred route, out of the routes assigned to r in the oscillation.

Since we assumed that r_m gained from its manipulation we deduce that:

$$v_{r_m}(T_{r_m}) < v_{r_m}(M_{r_m}) \tag{3.1}$$

Because r_m strictly prefers M_{r_m} to T_{r_m}, but did not choose it in the routing tree T, we must conclude that the route M_{r_m} is not available to r_m in T. This means that there must exist some node r (other than r_m) that is on the route M_{r_m} and that does not have the same route in M as it has in T. Let r_1 be the node on the path M_{r_m} that is closest to d on M_{r_m}, such that $M_{r_1} \neq T_{r_1}$.

By definition, all nodes that follow r_1 on the route M_{r_m} have exactly the same routes in T and in M. This means that the node r_1 could choose the route M_{r_1} in T. Since it did not choose that route we must conclude that:

$$v_{r_1}(M_{r_1}) < v_{r_1}(T_{r_1})^4 \tag{3.2}$$

[4] The reason that the inequality is strict is that equality can exist only if the two routes go through the same neighbouring node. This cannot be the case as $M_{r_1} \neq T_{r_1}$.

We can now proceed to the next step in the proof. Since T_{r_1} is preferred by r_1, and was not chosen by r_1 in the routing tree M, it must be that it was not an available option. Therefore, there is some node r on the route T_{r_1}, that is not r_1, such that $T_r \neq M_r$. We select r_2 to be the node r closest to d on the path T_{r_1} for which $T_r \neq M_r$. As before, all nodes closer to d than r_2 on the route T_{r_1} send traffic along identical routes in both T and M. Hence, the route T_{r_2} must be available to r_2 even in M. The fact that it was not chosen in M implies that r_2 prefers M_{r_2} over it. Thus, we have that:

$$v_{r_2}(T_{r_2}) < v_{r_2}(M_{r_2}) \tag{3.3}$$

We can continue these steps, alternating between the routing trees T and M and creating a sequence of nodes as follows:

- $r_0 = r_m$

for $n = 0, 1, 2, \ldots$ we perform the following steps:

- **M step**: Let r_{2n+1} be the node r on the route $M_{r_{2n}}$ such that $M_r \neq T_r$, and r is closest among all such nodes to d on $M_{r_{2n}}$.
- **T step**: Let r_{2n+2} be the node r on the route $T_{r_{2n+1}}$ such that $M_r \neq T_r$, and r is closest among all such nodes to d on $T_{r_{2n+1}}$.

Note that the destination node d cannot appear in this sequence because the route $L_d = T_d$ is the empty set. Due to our construction, and to arguments similar to the ones presented before, the preferences over routes are as follows:

$$\text{for } i = 0, 2, 4, \ldots \qquad v_{r_i}(T_{r_i}) < v_{r_i}(M_{r_i}) \tag{3.4}$$

$$\text{for } i = 1, 3, 5, \ldots \qquad v_{r_i}(M_{r_i}) < v_{r_i}(T_{r_i}) \tag{3.5}$$

Since there is only a finite number of nodes, at some point a node will appear in this sequence for the second time. We denote the first node that appears two times in the sequence by u_0. Let $u_0, ..., u_{k-1}, u_0$ be the subsequence of $r_0, r_1, ...$ that begins in the first appearance of u_0 and ends in its second appearance. We shall examine two distinct cased.

CASE I: The manipulator r_m does not appear in the subsequence $u_0, ..., u_{k-1}, u_0$.

Proposition 3.1 *If for all $i \in \{0, ..., k-1\}$ $r_m \neq u_i$ (the manipulator is not one of the nodes in the subsequence) then k must be even.*

Proof: If k is odd, then it must be that u_{k-1} and u_0 (in its second appearance in the subsequence) were both selected in M steps, or were both selected in T steps. However, if this is the case we reach a contradiction as both nodes were supposed to be the node r closest to d on a certain route, such that $T_r \neq M_r$. Since $u_{k-1} \neq u_0$ this cannot be. ∎

If k is even then the subsequence of nodes $u_0, ..., u_{k-1}, u_0$, along with the T_{u_i} and M_{u_i} route, and the preferences over these routes (expressed before) form a Dispute Wheel (as in Figure 3.5).

CASE II: The manipulator r_m appears in the subsequence (that is, $u_0 = r_m$). We now need to handle two subcases: The subcase in which k is even and the subcase in which k is odd. If k is even then the second appearance of the manipulator (u_0) in the subsequence is due to a T step. If so, a Dispute Wheel is formed, as in the example in Figure 3.5[5].

We are left with the subcase in which k is odd. In this case the second appearance of r_m was chosen in an M step. If so, it must be that $M_{u_{k-1}}$ (that goes through r_m) is not used in practice (otherwise, both u_{k-1} and that the second appearance of $r_m = u_0$ was chosen in M steps, and arguments similar to those of Proposition 3.1 would result in a contradiction). This must be the result of a manipulation by r_m. Let L_{r_m} be the false route reported by the manipulator to the node that comes before it on $M_{u_{k-1}}$. Recall that the manipulator can only announce a route L_{r_m} that exists and is available to it in M. Recall, that the second appearance of the manipulator was chosen due to an M step. Therefore, all nodes that follow it on $M_{u_{k-1}}$ (which are the same nodes as in L_{r_m} are assigned the same routes in T and M. Hence, L_{r_m} was available to r_m in T. It must be that $v_{r_m}(L_{r_m}) \leq v_{r_m}(T_{r_m})$, for otherwise r_m would have chosen L_{r_m} as its route in T (a contradiction to the stability of T). We know that $v_{r_m}(T_{r_m}) \leq v_{r_m}(M_{r_m})$ because we assumed that the manipulation performed by r_m was beneficial to it. We get:

$$v_{r_m}(L_{r_m}) \leq v_{r_m}(T_{r_m}) \leq v_{r_m}(M_{r_m}) \tag{3.6}$$

Thus, we form a Dispute Wheel with L_{r_m} as shown in Figure 3.6.

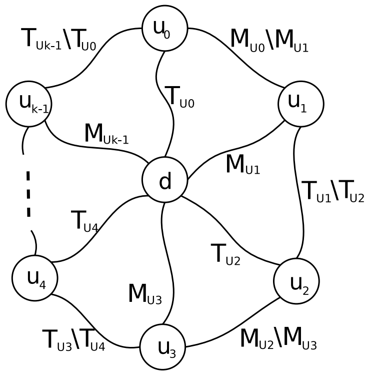


Figure 3.5: A Dispute Wheel constructed during the proof of Theorem 3.4 (even number of nodes).

∎

3.3.3 BGP With Route Verification Is Collusion-Proof

A similar, more complex, construction shows that Theorem 3.4 can be strengthened as follows:

Theorem 3.5 *If No Dispute Wheel and Route Verification hold, then the best-reply dynamics are collusion-proof in (subgame perfect) ex-post Nash.*

Proof: We shall assume, by contradiction, that a group of manipulators colludes in an interdomain routing instance with no Dispute Wheel and improve their routing outcomes. We define T_r and

[5] The route $R \backslash S$ (where S is a sub-route of R) is route R truncated before the beginning of S

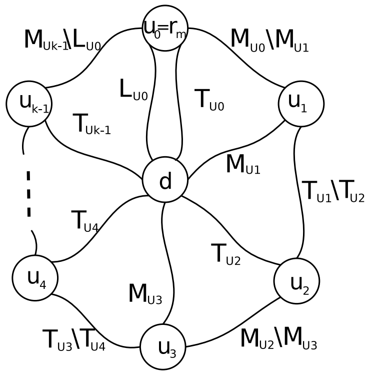


Figure 3.6: A Dispute Wheel constructed during the proof of Theorem 3.4 (odd number of nodes).

M_r as in the proof of Theorem 3.4. We assume, by contradiction, that all manipulators are not harmed by this manipulation:

$$\forall r \in Manipulators \quad v_r(T_r) \leq v_r(M_r) \tag{3.7}$$

We shall arrive at a contradiction by showing the existence of a Dispute Wheel in a similar manner to that demonstrated in the proof of Theorem 3.4.

We begin the construction by selecting one of the manipulators that strictly gained from the collusion. We shall denote this manipulator by r_m (it must be that $T_{r_m} \neq M_{r_m}$). We now construct a sequence of nodes in the following way (This is a slightly modified version of the construction appearing in Theorem 3.4):

- $r_0 = r_m$
- **M-manip step**: For a node r_n which is a manipulator we define r_{n+1} to be the node r on the route M_{r_n}, such that $M_r \neq T_r$, and r is the closest to d on M_{r_n} among all such nodes.
- **T step**: For a node r_n that is not a manipulator, and was chosen in an M step, we define r_{n+1} to be the node r on the route T_{r_n}, such that $M_r \neq T_r$, and r is the closest to d on T_{r_n} among all such nodes.
- **M-honest step**: For a node r_n that is not a manipulator, and was chosen in a T step, we define r_{n+1} to be the node r on M_{r_n}, such that $M_r \neq T_r$, and r is the closest to d on M_{r_n} among all such nodes.

The nodes in the sequence $r_0, r_1, r_2, \ldots$ must start repeating at some point. We denote by u_0 the first node that is the first to appear twice in the sequence, and by $u_0, ...u_{k-1}, u_0$ the sub-sequence of nodes between those repetitions. As in Theorem 3.4, we will reach a contradiction by showing that a Dispute Wheel exists in the graph. The nodes $u_0, ..., u_{k-1}, u_0$ will be the pivot-nodes. First, we prove the following proposition that will aid our construction.

Proposition 3.2 *There is no $i \in \{0, ..., k-1\}$ such that both u_i and u_{i+1} (modulo k) are manipulators (no two manipulators come one after the other in the subsequence $u_0, ...u_{k-1}, u_0$).*

Proof: By contradiction, let u_i and u_{i+1} be two consecutive manipulators. u_{i+1} was chosen in an M step. u_{i+1} is therefore the node r closest to d on M_{u_i} such that $M_r \neq T_r$. Hence, $M_{u_{i+1}}$ must be available to u_{i+1} in both M and T. We know that $v_{u_{i+1}}(T_{u_{i+1}}) \leq v_{u_{i+1}}(M_{u_{i+1}})$, as u_{i+1} is a manipulator. Since u_{i+1} chose $T_{u_{i+1}}$ over $M_{u_{i+1}}$ in T it must also be that $v_{u_{i+1}}(T_{u_{i+1}}) \geq v_{u_{i+1}}(M_{u_{i+1}})$. We conclude that $v_{u_{i+1}}(T_{u_{i+1}}) = v_{u_{i+1}}(M_{u_{i+1}})$. However, equality of the values of routes assigned by u_{i+1} is only possible if u_{i+1} forwards traffic to the same node in both routes. Since both routes are available in T, this means that $T_{u_{i+1}} = M_{u_{i+1}}$. This contradicts the reason for which u_{i+1} was selected ($M_{u_{i+1}} \neq T_{u_{i+1}}$). ∎

We will now need to prove that each node u_i in the (cyclic) sequence has two paths to its destination, a direct path Q_i, and one that passes through u_{i+1} that we will denote as R_iQ_{i+1}, and that each node prefers the indirect path over the direct one. We will examine three possible cases that match the three types of steps in the selection of the series of nodes.

- **M-manip step**: For a node u_i which is a manipulator we select the route R_iQ_{i+1} as the indirect route M_{u_i} that passes through u_{i+1}. This path must exist since it is the one the manipulator uses in the manipulation (and gains from its use).

  From proposition 3.2, we know that the node u_{i-1} is not a manipulator, since during the construction we picked node u_i after node u_{i-1}, and because honest nodes use Route Verification, it must be that the path from u_{i-1} to the destination through node u_i exists. We set the path Q_i to be the suffix of the path node u_{i-1} *believes* it is using. Node u_i was selected as the last node on this path that routes differently in M and T, and so this path must be available to node u_i in T. From this we learn that

  $$v_{u_i}(Q_i) \leq v_{u_i}(T_{u_i}) < v_{u_i}(M_{u_i}) = v_{u_i}(R_iQ_{i+1}) \tag{3.8}$$

  which is the preference relation we require to construct a Dispute Wheel.

- **T step**: For a node u_i that is not a manipulator, and was chosen in an M step, we define Q_i to be the path that this node has in M. We define its indirect route R_iQ_{i+1} as its path in T which must exist in the network, as it is used when all nodes play honestly, and which by construction passes through u_{i+1}. Note, that node u_i is the closest node to d on $M_{u_{i-1}}$, and all following nodes on this path route the same in M, T. However, this path is not chosen by u_i in T which means it is less preferred:

  $$v_{u_i}(Q_i) < v_{u_i}(R_iQ_{i+1}) \tag{3.9}$$

- **M-honest step**: For a node u_i that is not a manipulator, and was chosen in a T step, we define Q_i to be the path that this node has in T. We define its indirect route R_iQ_{i+1} as the path it *believes* it has in M (note that here we are consistent with the definition of Q_{i+1} for the case in which node u_{i+1} is a manipulator). This path must exist in the network due to Route Verification and by construction passes through node u_{i+1}. Note, that node u_i is the closest node to d on $T_{u_{i-1}}$, and all following nodes on this path route the same in M, T. However, this path is not chosen by u_i in M which means it is less preferred:

$$v_{u_i}(Q_i) < v_{u_i}(R_i Q_{i+1}) \tag{3.10}$$

The reader may check that the direct path defined as Q_{i+1} in each step is consistent with the suffix of $R_i Q_{i+1}$ that is defined in the previous step. We have therefore completed our construction of a Dispute Wheel from the topology of the graph, and preferences of nodes. This concludes the proof of the theorem. ∎

3.4 Maximizing Social Welfare

We prove that obtaining an approximation ratio better than n to the optimal social welfare is hard even if No Dispute Wheel holds. In fact, this can be shown even for the Gao-Rexford setting. We present two lower bounds, one in the computational complexity model, and one in the communication complexity model.

Theorem 3.6 *Obtaining an approximation of $O(n^{\frac{1}{2}-\epsilon})$ to the social welfare is impossible unless $P = NP$. Obtaining an approximation of $O(n^{1-\epsilon})$ to the social welfare is impossible unless $P = ZPP$. This holds for any $\epsilon > 0$ and even in the Gao-Rexford setting.*

Proof: Our proof will be by reduction from Clique. Assume a graph $G =< V, E >$, we construct a network with N nodes and L links. In this network, N consists of $2|V| + 1$ source-nodes and a unique destination node d. The source nodes are divided into 3 disjoint sets: Two sets N_1, N_2, such that $|N_1| = |N_2| = |V|$ and a *connection node c*. We associate a node $v(N_1) \in N_1$ and a node $v(N_2) \in N_2$ with every node $v \in V$. All nodes in N_1 are connected to the connection node c. All nodes in N_2 are connected to each other, to the connection node, and to d. See Figure 3.7.

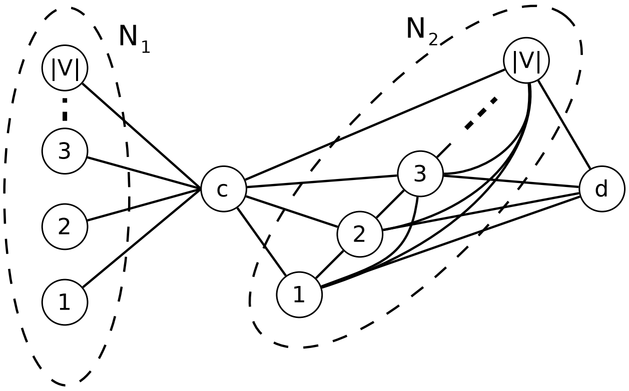


Figure 3.7: The network in the proof of Theorem 3.6

All nodes in N_2, and c, have valuation functions that assigns a value close to 0 to all routes (no ties). Fix some order O on the nodes in N_2. A node $v(N_1) \in N_1$ assigns a value close to 1 (no ties) to a route R iff *all* the following conditions hold:

- $(v(N_1), c)$ is the first link on R.

- The order of appearance of the nodes from N_2 in R is consistent with O.
- $v(N_2)$ is on R.
- For every node $u(N_2) \neq v(N_2) \in N_2$ on R there is an edge (v, u) in G.

Observe that since c is the only connection between N_1 and $N_2 \cup \{d\}$, c's route determines the routes of all nodes in N_1. The reader can verify that every clique in G corresponds to a routing tree with a social welfare that equals (almost) the size of the clique (assign every node in N_1 that is in the clique the route that goes through c and then the all the nodes in N_2 that are associated with nodes in the clique). In addition, the social welfare of every routing tree corresponds to a clique in the original graph G (the route from c to d through N_2 determines the identity of the clique). The theorem follows from the known inapprroximability results for Clique [53].

We note that this result can be made to hold in the Gao-Rexford setting by defining business relationships as follows: c is a customer of all nodes in N_1. For every two nodes $r, s \in N_2$, such that s comes after r in O, s is r's customer. d is a customer of all nodes in N_2. ∎

Theorem 3.7 *Obtaining an approximation of $O(n^{1-\epsilon})$ to the social welfare requires exponential communication (in n). This holds for any $\epsilon > 0$ and even in the Gao-Rexford setting.*

Proof: The proof is similar to the proof of Theorem 3.1, and is by reduction from Set Disjointeness. There are n parties, and each party i holds a subset A_i of $1, ..., K$. The goal is to distinguish between the two following extreme subcases:

- $\bigcap_i A_i \neq \emptyset$
- For every $i \neq j$ $A_i \cap A_j = \emptyset$

It is known that in order to distinguish between these two subcases the parties must exchange $\Omega(K)$ bits. We set $K = 2^{\frac{n}{2}}$.

Now, consider a network with $2n + 1$ source nodes and a unique destination node d. The set of nodes N consists of 3 disjoint subsets: n *sending nodes*, a *connecting node* c, and n *transit nodes*. Each party $i \in [n]$ in the Set Disjointeness problem is associated with a sending node s_i.

The transit nodes are divided into $\frac{n}{2}$ pairs $T_1, ..., T_{\frac{n}{2}}$. All sending nodes are connected to the connecting node, which, in turn, is connected to both nodes in T_1. For every $r = 1, ..., \frac{n}{2} - 1$, each node in T_r is connected to both nodes in T_{r+1}. Both nodes in $T_{\frac{n}{2}}$ are connected directly to d. See a description of the network in Figure 3.8.

All transit nodes and the connecting node have a value close to 0 (no ties) for all routes. There are $2^{\frac{n}{2}}$ possible routes from c to d that go through the transit nodes (see proof of Theorem 3.1). Fix an arbitrary order on these routes $R_1, ..., R_{2^{\frac{n}{2}}}$. The valuation function of each s_i assigns a value close to 1 (no ties) to a route $(s_i, c)R_a$ iff $a \in A_i$. It assigns a value close to 0 to all other routes (no ties). The reader can verify that there is a route assignment with a social welfare-value close to n if there is some $a \in \bigcap_i A_i \neq \emptyset$ (assign every node i the route $(s_i, c)R_j$). If, on the other hand, for every $i \neq j$ $A_i \cap A_j = \emptyset$, then any route assignment cannot have a social-welfare value better than $1 + \epsilon$ (this is because c's route determines the routes of the sending nodes through the transit nodes).

Hence, we have a network with $O(n)$ nodes, in which determining whether the social-welfare is n or 1 is equivalent to solving the Set Disjointeness problem with n players (each player i

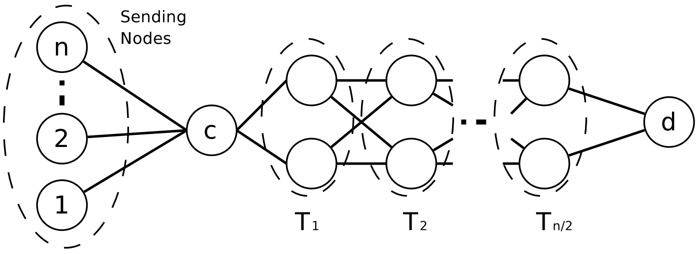


Figure 3.8: The network in the proof of Theorem 3.7

simulates node s_i), each holding a subset of $1, ..., 2^{\frac{n}{2}}$. It therefore requires at least $\Omega(2^{\frac{n}{2}})$ bits of communication. This concludes the proof of the theorem.

This result too can be made to hold for the Gao-Rexford setting if we define business relationships as follows: For every $i = 2, .., \frac{n}{2}$, the nodes in T_i are customers of the nodes in T_{i-1}. d is a customer of the nodes in $T_{\frac{n}{2}}$, the nodes in T_1 are customers of c, and c is a customer of all nodes in N_1. ∎

Remark 3.1 *As in the case of Theorem 3.1, the proof of Theorem 3.7 can easily be modified to show that even if we assume that nodes are only interested in routes of length at most x, obtaining an approximation of $O(x^{1-\epsilon})$ requires communicating $O(2^x)$ bits.*

A trivial upper bound of n can be achieved by finding the node with the highest value for some route, assigning that route to that node (thus getting an n-approximation), and then assigning routes to all other nodes in a way that forms a tree rooted in d.

3.5 Open Questions

- No Dispute Wheel is sufficient, but not necessary, for guaranteeing BGP convergence. Recall, that when we say BGP convergence we are referring to BGP convergence for *all* possible timings of update messages, a property known as "*BGP safety*" in networking literature [49]. Very recently (subsequently to our work), Sami et al. [88] presented a more general sufficient condition for BGP convergence than No Dispute Wheel called, "*Iterated Dominance Tree*". In the previous chapter we have shown that the existence of two or more stable solutions always implies that BGP safety does not hold, thus providing the first non-trivial *necessary* condition for BGP convergence (uniqueness of the stable solution). However, there is still a big gap between the known sufficient conditions, and this new necessary condition. **An important open question if to find non-trivial *characterizations* (sufficient and necessary conditions) of networks for which BGP safety is guaranteed.**

- **What more general conditions than No Dispute Wheel and Route Verification guarantee incentive-compatible convergence of best-reply dynamics in our interdomain routing model?** One can try to identify such more general conditions by relaxing the No Dispute Wheel assumption, and/or the Route Verification requirement. Indeed, Sami et al. [88] have been able to strengthen our results, by showing that similar results hold for a condition broader than No Dispute Wheel (called Iterated Dominance Tree), and a security requirement weaker than Route Verification (called Topology Validation). However, we still do not have a good understanding of which conditions imply incentive-compatibility.

No Dispute Wheel does not only guarantee BGP convergence, but even *robust BGP convergence.* That is, not only is BGP guaranteed to converge, but it will converge even if nodes and links are arbitrarily removed from the network (because for every subgraph of the AS graph No Dispute Wheel holds). This is very important from a practical perspective as it implies that BGP convergence is achieved even in the presence of link and node failures. We present the following conjecture:

Conjecture: *Robust BGP convergence always implies incentive-compatibility of BGP with Route Verification.*

That is, whenever the network and routing policies are such that robust BGP convergence is guaranteed, BGP with Route Verification is incentive-compatible.

- **Other Models.** The questions we consider in this chapter, and our formal framework, can be applied to variations of our model. For instance, subsequently to our work Goldberg et al. [45] have presented several impossibility results for the case that utility functions in our model are also influenced by incoming traffic (i.e, nodes do not only care about the routes they get, but also about which nodes are routing through them). Similarly, Kintali [58] extended our game-theoretic model to the case of Fractional BGP [54]. It will be interesting to study incentive-compatibility of best-reply dynamics in various other interdomain routing settings.

- **Identify realistic interdomain routing settings in which good approximations to the social-welfare are possible.** It has been shown [37] that No Dispute Wheel and Policy Consistency imply that BGP always converges to the socially-optimal routing tree. It was also shown [37] that relaxations of Policy Consistency lead to arbitrarily bad solutions, even in very small and realistic networks. We are interested in identifying settings in which BGP is guaranteed to reach solutions that have a good social-welfare value. In our game-theoretic model, this question translates to a *price of anarchy* [59] problem: We wish to identify restrictions on the topology of the network, and the routing policies of nodes, for which the worst-case ratio (with respect to social welfare) between pure Nash equilibria in the One-Round Game (stable solutions), and the optimal solution, is small.

  There is also the computational question: Our results in this chapter show that even in the Gao-Rexford setting one cannot find a good approximation to the optimal social welfare (via *any* algorithm) *in polynomial time.* In which settings is it possible to obtain a good approximation in polynomial time (via BGP, or another protocol)?

Chapter 4

Best-Reply Mechanisms

4.1 Introduction

In Chapters 2 and 3 we have studied the properties of the Border Gateway Protocol (BGP). We observed that BGP can actually be regarded as best-reply dynamics in a complex routing game. Indeed, in many real-world settings, strategic agents are instructed to follow best-reply dynamics (economic markets, Internet protocols, etc.). Such settings give rise to a natural question: Is it in the best interest of the agents to follow best-reply dynamics? That is, are best-reply dynamics incentive-compatible? (As we have seen, incentive-compatibility is an important requirement from protocols like BGP.) To answer this question, we initiate the study of best-reply dynamics in the context of mechanism design.

We present a formal framework for the strategic analysis of best-reply dynamics, and exhibit several interesting examples in which best-reply dynamics are incentive-compatible: adword auctions, congestion control settings inspired by the Transmission Control Protocol (TCP), cost-sharing problems, economic stable matching problems, and interdomain routing environments in which best-reply dynamics correspond to the Border Gateway Protocol (BGP).

4.1.1 Best-Reply Mechanisms

The most natural approach for finding a (pure) Nash-equilibrium of a given game is executing "*best reply dynamics*": Start with an arbitrary strategy profile of the players, and in each step let some player switch his strategy to be the best reply to the current strategies of the others. If this process converges, then we have reached a pure Nash equilibrium, and moreover, a "highly justifiable" one. The many qualities of best-reply dynamics (simplicity, distributed nature, low costs in terms of memory, and many more) make it very attractive from an algorithmic perspective. Indeed, many protocols used in practice are essentially executions of best-reply dynamics. As we have seen in Chapter 3, a noteworthy example for this is the *Border Gateway Protocol (BGP)*, that handles *interdomain routing* in today's Internet [65]. Other real-life settings in which best-reply dynamics plays a crucial role are economic markets.

Best-reply dynamics is the subject of a large body of literature in game-theory and computer-science. Much work has been devoted to questions like "*When is best-reply dynamics guaranteed to converge to a pure Nash equilibrium*?" [85, 68], and "*What is the convergence-rate of best-reply dynamics*?" [27, 26, 1]. In this Chapter we present and address a natural question that has received little, if any, attention: "*When is following best-reply dynamics the best course of action for every player?*". In interdomain routing, for example, we wish to make sure that the computational nodes cannot better their routing outcomes by "deviating" from BGP [65]. Intuitively, we want best-reply dynamics to be in some sort of equilibrium, i.e, as long as other players are continuously best-replying it should be in the best interest of a player to do the same. This notion of game dynamics that are in equilibrium is related to the notion of "learning equilibrium" in repeated games, put forth by Brafman and Tennenholtz in [13, 14].

We consider best-reply dynamics in the context of *mechanism design.* An appealing property of best-reply dynamics is that the actions of players are "*uncoupled*": each player's best-reply depends on his own utility function and not on the utilities of other players. Hence, best-reply dynamics are feasible strategies in games with private information. Often, in such contexts, best-replying is adopted without any strategic justification ("myopic play"). We wish to identify situations where best-replying is strategically justified – i.e. *incentive compatible.*

Our framework is the standard one of mechanism design with private values, where each player's private value is his own utility function. In many such scenarios a natural objective is to implement a Nash equilibrium of the full-information game induced by players' utilities. An indirect mechanism that we may hope can implement this outcome is a *"repeated-reply mechanism"* where players repeatedly announce a reply to other players' most recent announcements. We refer to repeated-reply mechanisms in which the prescribed behaviour for players in best-replying as "best-reply mechanisms". Unfortunately, we observe that best-replying in this mechanism is not necessarily incentive-compatible, even in very simple games.

However, best-reply mechanisms turn out to be a generic and extremely useful tool for the design of *computationally-efficient* and incentive-compatible algorithms in various important settings. We point out several realistic environments in which best-reply dynamics is incentive-compatible. For all of the examples below, we prove (1) convergence of best-reply dynamics to a pure Nash equilibrium, (2) that best-reply strategies are incentive compatible in the ex-post-Nash sense. In fact, we prove that for many of these cases best-reply dynamics has many other useful properties, like convergence to a Pareto optimal state, incentive-compatibility even in asynchronous settings, and resilience even to manipulations by coalitions of players of any size (*collusion-proofness*). This gives broader context, strengthens, and unifies several known results, and leads to many new ones.

- **Iterative Single-Item Auctions:** This simple example, mainly used to illustrate our framework, deals with a classic economic setting. A single item is sold in a first price auction (with discretized bids). The repeated setting allows players to repeatedly change their bid in response to the other bids, until no further changes are desired. We show that in this fundamental setting best-reply dynamics (with given tie-breaking rules) mimic, in certain cases, the behavior of the well-known ascending English auction. Notably, in our framework the ascending behavior is completely endogenous and not dictated in any way by the auction rules. Hence, we present a new dynamic incentive-compatible auction that generalizes the English auction.

- **Adword Auctions:** Sponsored search advertising is a major source of revenue for Internet search engines (close to 98 percent of Googles total revenue of 6 billion for the year 2005 came from sponsored search advertisements). In the adword auctions setting [25, 99] there are k slots that are to be sold to different advertisers. Each slot has a clique-through-rate (CTR) (higher slots are preferred to lower ones). Each bidder (advertiser) has a private value

v_i per click. The auction rule used by companies like Google and Yahoo! is the *generalized second-price auction* (GSP). GSP assigns the k slots to the k highest bidders (the highest bidder gets the highest slot, etc.) and charges each winning bidder a cost per-click that equals the bid of the bidder that was assigned the slot below his.

Adword auctions are dynamic in nature – the advertisers are allowed to change their bids quite frequently. It is known that a generalized version of the ascending English auction always reaches a specific, economically reasonable, allocation of slots in an incentive-compatible manner [25]. As in the single item case, the generalized English auction can be regarded as a special case of best-reply dynamics. However, the more general (and arguably more realistic) case is much more problematic; it was shown by Cary et al. [16] that best-reply dynamics in this setting are not even always guaranteed to converge. However, [16] shows that this problem can be overcome by using randomness. We prove that this randomized convergence of best-reply dynamics is also incentive-compatible.

- **Congestion-Control Games:** These games are inspired by the Transmission Control Protocol (TCP).[1] The setting consists of a directed graph, with the nodes being non-strategic routers. Strategic players ("flows") want to transmit packets along fixed paths in the network. The routers drop packets if the network is congested, using Weighted Fair Queuing [20]. Flows must decide on their transmission rates, and wish to maximize their throughput. We consider a natural distributed congestion-control protocol that simulates best-reply dynamics. We show that this protocol reaches a stable capacity-shares outcome and is incentive-compatible even in asynchronous Internet-like settings. Furthermore, the obtained Nash-equilibrium satisfies max-min fairness.

- **Cost-Sharing Games:** Cost-sharing problems arise in situations in which we wish to distribute the cost of some public service (e.g., building a bridge) between self-interested users that will benefit from this service in different extents. A cost-sharing problem is defined by a cost function C, that specifies, for every subset of users, the cost of providing the public service to these users. Every user is assumed to have a private value for being serviced. A cost-sharing mechanism outputs, for every possible combination of private values of the users, a set of users that may benefit from the public good, and the way that the cost will be divided between these users. A well-known family of truthful cost-sharing mechanisms are the Moulin mechanisms [70, 71]. We show that such mechanisms can be implemented in a *distributed*, incentive-compatible manner (by the users themselves) via best-reply dynamics. In [66], a more general family of truthful cost-sharing mechanisms, called "acyclic mechanisms" is presented. These mechanisms have various advantages over Moulin mechanisms. In particular, they follow in a generic way from off-the-shelf primal-dual algorithms. Our results for Moulin mechanisms can be extended to this more general class.

- **Stable-Roommates Games:**

  This well known example [41] considers how to pair students (say, for the purpose of sharing a dorm room), when each student has his own preference order over possible roommates. The classic goal is to find a "stable matching" where no two students prefer each other over their assigned roommates. A natural mechanism stipulates that students repeatedly propose

[1] The study of TCP from a game-theory/mechanism design perspective is a long-standing agenda [81, 40, 3].

to their most preferred roommate among those that do not immediately reject them, and immediately reject all proposers except for their most preferred one so far. This may be formalized as a game, and for the two well-known cases described below we prove that best-reply dynamics converge to a stable matching in an incentive-compatible manner[2].

  - **Intern-Assignment Games:** This example considers how to assign interns to hospitals. Each intern has his own preference order over possible hospitals. The hospitals have a common way to rank interns (say, according to their GPA).
  - **Correlated Two-Sided Markets:** These games are inspired by real-life one-sided market games in which players have preferences about a set of markets, and the preferences of markets are correlated with the preferences of players [1] (e.g., market sharing games, and distributed caching games). The game is represented by a weighted bipartite graph in which vertices on one side represent players (buyers), and vertices on the other side represent markets (sellers). Each buyer has a preference order over sellers, such that a seller A is ranked higher than another seller B by that buyer if the edge connecting that buyer to A has a bigger weight than the edge connecting the buyer to B. Similarly, sellers prefer buyers that are connected to them by heavier edges.

- **Routing Games:** These games, presented in Chapter 3, are a model of interdomain routing in the Internet, where best-reply dynamics model the Border Gateway Protocol (BGP). The setting assumes strategic nodes in a directed graph, each wishing to establish a path in the network from itself to a given fixed target node d. Each node has its own preference order over all possible paths from itself to d (a preference that may depend on various network parameters, as well as on various business parameters). Each node's action is choosing a single outgoing edge. The collection of these outgoing edges determines a unique path (or none) from each node to d. The networking community has spent considerable effort into understanding the convergence of BGP and has come up with the well-known Gao-Rexford commercial constraints [43]. We show that under these constraints BGP is incentive-compatible in realistic asynchronous settings, thus placing the recent result of Levin et al. [65] in a broader context.

From a technical perspective, many of our results are based on an identification of a subclass of games, called "universally-max-solvable" games, for which we prove that best-reply dynamics is incentive-compatible (and has even significantly stronger properties). The main challenge in proving our results is therefore showing that the games in question indeed belong to this subclass (which is achieved by proving that they have a specific combinatorial structure). Universally-max-solvable games are very restricted, seemingly almost empty. However, we show that they are expressive enough to contain many well-studied examples. In the case of games that do not fall into the restricted category of universally-max-solvable games, like adword auctions, we need to resort to different techniques. In particular, we discuss more general (than universal-max-solvability) structural properties that games can possess for best-reply dynamics to be incentive-compatible. We prove that these properties hold for adword auctions.

[2] Unlike in the related stable-marriage problem [41], in this unisex variant it is known that stable matchings need not exist in general.

4.1.2 Organization of the Chapter

In Section 4.2 we present our formal framework and illustrate it via the iterative single-item auctions exmaple discussed above. In Section 4.3 we exhibit several best-reply mechanisms for well-known realistic settings. In Section 4.4 we prove the results presented in Section 4.3 and discuss our proof techniques.

4.2 Best-Reply Dynamics and Mechanism Design

In this section we present our formal framework of best-reply mechanisms, and illustrate it via a simple auction setting. We then explain how our framework can be extended to incorporate randomness, and to handle realistic asynchronous environments.

4.2.1 Best-Reply Mechanisms

Let G be a *game with private information*, in which players have *action sets* $A_1, \dots, A_n$. Each player i has a utility function u_i that assigns a value to each action-profile, and is his private information. $u = (u_1, \dots, u_n)$ belongs to a set of utility-function profiles U. An action a_i is said to be a best-reply of i to the actions of the other players a_{-i} if it maximizes u_i, given a_{-i}.

Our results will be that, for certain private-information games, best-replying implements is incentive compatible. This turns out to be true in quite general settings, but for clarity of presentation we will start by considering a simple centralized and synchronous model. Our model can be extended to distributed settings (with no computational centre) with enough rounds of interaction (even asynchronous) as well as to settings where non-convergence is (or can be made) undesired by the players (e.g., just because of time-value). The only significant properties of this model for our results are:

1. That there be a sufficient number of "rounds" for best-reply dynamics to converge to a pure Nash equilibrium.

2. Ensuring that "non-convergence" is not in the best interest of any player.

Our model captures the first point by simply considering a synchronous round-robin protocol, and the second point by enforcing a fixed "penalty" outcome if convergence is not reached by a predefined deadline.

Definition 4.1 *A penalty in a game with private-information G is an action-profile a such that for all $u = (u_1, ..., u_n) \in U$, for every player i, and for every action profile a', it holds that $u_i(a) \leq u_i(a')$.*

Definition 4.2 *A repeated-reply mechanism for a game with private information G, is an extensive form game with perfect recall, that has the same players as G, and consists of at most M steps. In each step of the game, a single player announces an element of A_i. Players play in round-robin order (for some fixed order on players). This goes on until all players "pass" in n consecutive steps (we say that a player "passed" in step i if he announced the same action in step i and in step $i-n$). In this case the mechanism enforces the the action profile of the most recently announced actions of the players. The payoff of a player in this case is his utility for this action-profile in G. If M steps go by and this does not occur, then the mechanism penalizes the players.*

Definition 4.3 *The best-reply strategy of player i in a repeated-reply mechanism (starting from an initial action profile $a = (a_1, \ldots, a_n)$) is to repeatedly announce the best-reply to the most recently announced actions of the other players (or to a in the first $n-1$ steps).*

Best-reply dynamics is the strategy-profile in which all players play best-reply strategies. Best-reply dynamics is said to be incentive-compatible in ex-post Nash if it is an ex-post Nash equilibrium [95] in a repeated-reply mechanism. Ex-post Nash in this case roughly means that if all other players are playing best-reply strategies, then a player cannot gain by not playing a best-reply strategy (this is true for *every* possible realization of the private-information of the players). Best-reply dynamics is collusion-proof in ex-post Nash [39] if no coalition of players of any size can deviate from it and strictly improve the outcome of one member in the coalition without strictly harming another. We shall refer to a repeated-reply mechanism as a "best-reply mechanism" if the prescribed strategy-profile [95] (that the players are instructed to follow) is best-reply dynamics.

4.2.2 Illustration: Iterative Single-Item Auction

The well-known ascending English auction can be viewed as a special case of a best-reply mechanism for the "1st price auction game":

The Game: A single item is sold in a first price auction. There are n bidders (players) and each bidder i has a private value v_i for the item and submits a single bid b_i from a pre-defined discrete set of bids (actions), say, the integers $\{0, 1, \ldots, K\}$. The winner is the bidder with the highest bid b_i (with ties broken, say, lexicoraphically) who pays his bid b_i for the item (and gets utility $v_i - b_i$.)

The Mechanism: Bidders repeatedly announce, one by one and in round-robin order, bids from the set of bids until all bidders "pass" (i.e., repeat their previous bid), at which point the final bids are used. If after $n^2K + n$ rounds this does not occur then the item is left unsold.

The Prescribed Best-Reply Strategies: In each step, bidder i bids the lowest bid that beats all other current bids, provided that this bid is feasible for him, i.e., less than v_i. If no feasible bid can beat the other current bids, he bids the largest feasible bid (i.e., at most v_i).

In Section 4.4 we prove the following theorem:

Theorem 4.1 *The prescribed best-reply strategies are incentive compatible and converge to the pure Nash equilibrium in which the highest bidder wins the item and is charged the (discretized) second highest bid.*

When the initial bids of the bidders are all 0, we get the familiar English auction behavior in which players repeatedly increase their bids by the minimal increment, and drop from the auction (i.e., pass) once their values are exceeded. However, note that our mechanism does not force ascending behavior, does not forbid jumps or stalling (passing and then increasing the bid), does not require 0 as the initial bid, and yet is guaranteed to end up with the same equilibrium.

4.2.3 Extensions of Best-Reply Mechanisms

Randomized Best-Reply Mechanisms

The definition of a randomized best-reply mechanism is almost identical to that of a best-reply mechanism (in Subsection 4.2.1). The only difference is that while in the definition of best-reply mechanisms players are chosen to play in round-robin order, in a randomized best-reply mechanism one randomly chosen player may update his action in each step.

Asynchronous Best-Reply Mechanisms

So far, our model considered best-reply dynamics in settings in which one can assume the presence of a central entity that synchronizes players' actions (make sure that players play in round-robin order, or choose one player to play in each step at random). However, in some of the settings considered in this chapter (inspired by the Internet) no such entity exists (asynchronous environments). In such settings, we can think of the players as residing in a computer network, where their best-replies are transmitted to other players and serve as the basis for the other players' best-replies (as is often the case in the Internet [48]). In this asynchronous environment, several players may choose their best-replies simultaneously. More generally, messages may be delayed or lost and players may choose their best-replies based on information that is out of date and unsynchronized with others.

To analyze best-reply dynamics in asynchronous settings, we change our definition of a repeated-reply mechanism: An *asynchronous repeated-reply mechanism* for a game with private information G, is still defined as an extensive form game with perfect recall, that has the same players as G. However, now the repeated-reply mechanism has an *infinite* number of steps. In each step one *or more* players are chosen to participate by an adversarial entity called the *Scheduler* [65]. A player i chosen to participate can make one or more of the following moves:

- Receive update messages from other players.
- Choose an element of A_i (this represents the action i actually chooses).
- Send an update message to another player announcing an element of A_i (this represents the action i announces to the other player).

The scheduler is in charge of making choosing the initial action profile of the players $a = (a_1, \ldots, a_n)$, determining which players participate in each step, and determining when sent update messages reach their destinations. The scheduler is restricted not to indefinitely starve any player from best-replying (that is, each player participates in infinitely many rounds). We name all choices made by the scheduler a *schedule.*

A strategy of a player in an asynchronous repeated-reply mechanism specifies the moves that a player makes in each step (as a function of his private information and the history of moves). The best-reply strategy of a player i is to repeatedly:

- Read update messages from all players.
- Choose the best-reply a_i to the most recent actions of the other players *that were announced to i.*
- Announce a_i to all other players.

The payoff of a player i in an asynchronous repeated-reply mechanism (given a specific schedule) is defined to be $u_i(a)$, where $a = (a_1, \ldots, a_n)$ is the profile of *actual* players' actions that appears in infinitely many rounds, and is assigned the highest value by u_i (this liberal definition of a player's payoff in the case that there was no convergence to a single action profile only strengthens our results).

4.3 Examples of Best-Reply Mechanisms

In this section we provide several examples of private-information games for which best-reply dynamics implements a pure Nash equilibrium (of the underlying full-information game) in an incentive-compatible manner: adword auctions, congestion control games, cost-sharing games, stable-roommates games, and routing games. The proofs appear in Section 4.4.

4.3.1 Adword Auctions

The Game. There are k slots $1, ..., k$. Each slot j has a clique-through-rate (CTR) α_j such that $\alpha_1 > \alpha_2 > \ldots > \alpha_k$ (we shall call the low-index slots "higher slots", as, in practice, higher slots are preferred to lower ones). There are n bidders. Each bidder i has a private value v_i per click. If bidder i is assigned slot j his gain is defined to be $\alpha_j \cdot v_i$. Each bidder submits a single bid b_i from a pre-defined discrete set of bids (multiplications of some arbitrarily small $\epsilon > 0$). The allocation rule assigns the k slots to the k highest bidders (the highest bidder gets slot 1, etc.) and charges each winning bidder a cost per click that equals the bid of the bidder that was assigned the slot below his. We denote by $\pi(j)$ the player that was awarded the j'th slot. That is, the utility of a player that is assigned to slot j is

$$u_{\pi(j)} = \alpha_j \cdot v_{\pi(j)} - \alpha_j \cdot b_{\pi(j+1)}$$

where $b_{\pi(j+1)}$ is the bid submitted by the bidder who is assigned the $(j+1)$'th slot.

The Mechanism: In each step choose one bidder, uniformly at random, and allow that bidder to change his bid. This goes on until all bidders do not change their bids (i.e., repeat their previous bids), at which point the final bids are used to allocate the slots. If after some large number of steps T (to be defined later) this does not occur then no slot is allocated.

The Prescribed Best-Reply Strategies: In each step, the bidder i chosen to play in that step chooses the bid b_i that maximizes his utility (given the bids of the other bidders and the allocation rule). Ties between different such bids will be broken as follows: A bidder i that bids so as to get slot j, should bid a value b_i such that

$$\alpha_j \cdot (v_i - b_{\pi(j+1)}) = \alpha_{j-1} \cdot (v_i - b_i)$$

That is, i chooses a bid b_i such that i is indifferent between getting the j'th slot and paying the next highest bid, or getting the $(j-1)$'th slot and paying b_j. If any best-reply bid of i leads to i not getting a slot at all, then i will bid his highest feasible bid, i.e, the maximal bid b_i that is smaller or equal to v_i.

Theorem 4.2 *The prescribed best-reply strategies are incentive-compatible and converge to a pure Nash equilibrium in which every bidder pays his VCG price.*

[16] have shown that the (randomized) best-reply dynamics indeed converges to this pure Nash equilibrium. We strengthen this result by showing that this is achieved in an incentive-compatible manner.

4.3.2 Congestion-Control Games

The Game. There is a directed graph $G =< V, E >$ in which the vertices represent non-strategic routers, and the edges represent communication links between them. There is a capacity function c that assigns every $e \in E$ a capacity $c(e) \in (0, 1]$. The players are *n flows.* Each flow is represented by a route r_i between a pair of nodes in V. Each flow i must choose a transmission rate $a_i \in [0, d_i]$, where d_i is a private value that represents i's maximal transmission rate.

The routers share the capacity of the edges between the flows in the following way: Let $e = (a, b)$ be an edge that appears on the routes of k flows. If the a_is of all k flows are at least $\frac{c(e)}{k}$ then each of the k flows will get an equal capacity-share of $\frac{c(e)}{k}$. Otherwise, the flow with the minimal transmission rate a_j gets a share of a_j and the remaining capacity of $c(e) - a_j$ is shared between the $k-1$ remaining flows in a similar manner (with $c(e)$ substituted by $c(e) - a_j$ and k substituted by $k - 1$). This corresponds to what is known as "Weighted Fair Queueing" in networking literature.

The *throughput* of a flow is the value of its minimal capacity-share, taken over all the edges its route traverses. Flows are selfish and wish to maximize their throughput.

The Mechanism: Flows repeatedly, one by one and in round-robin order, choose transmission rates. This goes on until all flows do not change their transmission rates (i.e., repeat their previous choices of transmission rates), at which point the final transmission-rates are used. If after $2n(n + |E|) + n$ rounds this does not occur then no flow is allowed to transmit at all.

The Prescribed Best-Reply Strategies: In each step, flow i chooses the *lowest* transmission-rate that maximizes its throughput (given the current transmission rates of the other flows).

Theorem 4.3 *The prescribed best-reply strategies are incentive compatible and converge to a pure Nash equilibrium for which the max-min fairness value is optimized. This holds even in asynchronous settings.*

4.3.3 Cost-Sharing Games

The Game. A cost-sharing problem is specified by a cost function C defined over a universe of users U. For every subset of the users S, $C(S)$ represents the total cost that players in S are charged if serviced. We assume that C is nonnegative ($C \geq 0$) and nondecreasing ($\forall S \subseteq T\ C(S) \leq C(T)$. There is a *cost-sharing method* ρ which specifies, for every subset of the users S, how the cost $C(S)$ will be shared between the users in S (each user $i \in S$ is assigned a nonnegative cost-share $\rho(i, S)$ such that $\Sigma_{i \in S} \rho(i, S) = C(S)$. Every user $i \in U$ has a private value v_i for being serviced, and submits a single bid b_i from a pre-defined discrete set of bids, say, the integers $\{0, 1, \ldots, K\}$.

For every combination of private values, the outcome of the game is defined in the following way: If for all $i \in U\ b_i \geq \rho(i, U)$ then all users are serviced and each user i is charged his bid b_i. If not, then the first user j (given some order $1, ..., n$) for whom $b_i < \rho(i, U)$ is removed and the same procedure is applied to the smaller universe $U \setminus \{j\}$. This recursion continues until a set of users whose bids exceed their cost-shares is reached, or no users are left (in which case no one is serviced).

The Mechanism: Users repeatedly, one by one and in round-robin order, submit bids. This goes on until all users pass (i.e., repeat their previous bids), at which point the final bids are used. If after nK rounds this does not occur then no user is serviced.

The Prescribed Best-Reply Strategies: In each step, user i bids the lowest bid that is a best-reply to the current bids of the other users.

Definition 4.4 *[70, 71] A cost-sharing method ρ is said to be cross-monotonic if $\forall i \in S \subseteq T$ $\rho(i, S) \geq \rho(i, T)$.*

Theorem 4.4 *For every cross-monotonic cost-sharing method, the prescribed best-reply strategies are incentive compatible.*

Our result shows that the famous Moulin mechanisms [70, 71] can be implemented in an incentive-compatible distributed fashion via best-reply dynamics. These results can be extended to the more general class of acyclic mechanisms [66].

4.3.4 Stable-Roommates Games

The Game. [41] There are n students $1, \ldots, n$ who must be paired to dorm rooms. Each student has a private full order of *strict* preferences over the other students. Each student must choose another student to be his roommate. Students are selfish and wish to be paired with students that are ranked as high as possible (given their preferences).

A stable matching is a matching of students such that no pair of students that are not matched to each other prefer each other to their matched roommates.

The Mechanism. We go over the students in round-robin order. At each step a single student announces another student. We call this an offer made by the student to the other student. This goes on until all students do not change their offers (i.e., repeat their previous roommate choices), at which point every two students that made an offer to each other (in their final announcements) are matched, and all other students remain unmatched. If after $n^3 - 2n^2 + n$ rounds this does not occur then all students remain unmatched.

The Prescribed Best-Reply Strategies: If student i announces a student j, and j prefers i over all other offers made to it (in the most recent student announcements) we say that i makes j a better offer. In each step, a student must choose the student he prefers most out of all students to whom he can make a better offer.

We show the the prescribed best-reply strategies are incentive-compatible a converge to a unique stable matching in the following natural special cases:

Intern-Assignment Games

The n students consist of interns and hospitals. Each intern has a private full order of *strict* preferences over the hospitals. All hospitals have identical preferences over interns (not necessarily known to the interns).

Theorem 4.5 *For intern-assignment games the prescribed best-reply strategies are incentive compatible and converge to a unique stable matching.*

Correlated Two-Sided Markets

The students consist of sellers and buyers. The game is represented by a weighted bipartite graph (for simplicity, assume that no two edges have the same weight) in which vertices on one side represent buyers, and vertices on the other side represent sellers (markets). Each buyer has a preference order over sellers, such that a seller A is ranked higher than another seller B by that buyer if the edge connecting that buyer to A has a bigger weight than the edge connecting the buyer to B. Similarly, sellers prefer buyers that are connected to them by heavier edges.

Theorem 4.6 *For correlated two-sided markets the prescribed best-reply strategies are incentive compatible and converge to a unique stable matching.*

4.3.5 Routing Games

The Game. A *routing game* [65], is played over a graph G with vertex set V, and an edge set E. The players are the vertices, and each vertex has a private full order of *strict* preferences over all the simple paths in the graph between itself and a common fixed target node d. Each vertex must choose an outgoing edge representing a choice of neighbouring node to forward traffic to. Vertices are selfish and wish to be assigned as highly ranked routes as possible (given their preferences).

We consider the well-known *"No Dispute Wheel"* [48] condition, which is known to generalize the Gao-Rexford setting [43] (see [65] and references therein).

The Mechanism. We go over the vertices in round-robin order. In each step, a single vertex chooses an outgoing edge. This continues until all vertices do not switch their outgoing edges (i.e., repeat their previous edge-choices). Then, the outputted route assignment consists of the final outgoing edges. If after $|V|^3 - 2|V|^2 + |V|$ rounds this does not occur, then no vertex is assigned an outgoing edge (no traffic is sent at all).

The Prescribed Best-Reply Strategies: In each step, a vertex a should choose a single outgoing edge such that the route from a to d induced by this choice (given current choices of other vertices) is most preferred by a (if no such route is induced then a does not choose an outgoing edge at all).

Corollary 4.1 *If No Dispute Wheel holds, then the prescribed best-reply strategies are incentive-compatible and converge to a unique pure Nash equilibrium. This holds even in asynchronous settings.*

This sheds light on the recent results presented in Chapter 3.

4.4 Proof Techniques

In this section we explain how to prove the results presented in Section 4.3. We exhibit a subclass of games, called "universally-max-solvable" games in which best-reply dynamics is incentive-compatible (and even has significantly stronger properties). We prove that many of the examples presented in Section 4.3 fall within this category of games. In the case of our adword auctions result, we discuss a different proof technique.

4.4.1 Universally-Max-Solvable Games

Full-information games. We define a subclass of *full-information games* called *universally max-solvable games.*

Definition 4.5 *Let S_i be the set of player i's strategies. We say that a set $T \subset S_i$ is universally max-dominated if $T \neq S_i$, $T \neq \emptyset$, and $max_{s_{-i}, s_i \in T} u_i(s_i, s_{-i}) < min_{s_{-i}, s'_i \in S_i \setminus T} u_i(s'_i, s_{-i})$.*

Intuitively, a subset of player i's strategies is universally max-dominated if the maximal utility player i can derive from playing a strategy in it is less than the minimal utility he can derive from playing a strategy that is not in it.

Definition 4.6 *A game G is said to be universally max-solvable if there is a sequence of games $G_0, \ldots, G_r$ such that:*

- *$G_0 = G$*
- *For every $k \in \{0, \ldots, = r-1\}$, G_{k+1} is a subgame of G_k achieved by removing a universally max-dominated strategy-set from the strategy space of one player in G_k.*
- *The strategy space of each player in G_r is of size 1.*

We shall refer to an elimination order of universally-max-dominated strategy-sets, that results in a single strategy-profile as an *elimination sequence* of a universally-max-solvable game.

Definition 4.7 *A strategy profile $s^* = (s^*_1, \ldots, s^*_n)$ is said to be collusion-proof if the following holds: For any subset of players T, there is no pure strategy-profile $s = (s_1, \ldots, s_n)$ such that for all $i \notin T$ $s_i = s^*_i$, for some $j \in T$ $u_i(s) > u_i(s^*)$, and for all $i \neq j \in T$ $u_i(s) \geq u_i(s^*)$.*

Proposition 4.1 *A universally max-solvable game has a unique pure Nash equilibrium. This pure Nash equilibrium is collusion-proof.*

Proof: By contradiction. Fix an elimination sequence of universally-max-dominated strategy-sets for the game (by definition, one exists). Let s^* be the strategy-profile reached after eliminating all the universally-max-dominated strategies in the elimination sequence. Let s be a strategy-profile such that s can be reached from s^* by changing the strategies of of some subset of the players T, for every $i \in T$ $u_i(s) \geq u_i(s^*)$ and for at least one player in T this inequality is strict. Let s_i be the strategy in s that is eliminated first (given this elimination sequence) of a player $i \in T$. Consider the elimination step in which s_i is eliminated. The strategy-set containing s_i is universally-max-dominated. This means that any strategy outside this set *always* guarantees i a *strictly* higher utility (given that all previous strategy-sets in the elimination sequence are removed). In particular, this is true for s^*_i. However, observe that both s and s^* do not contain any strategies that were removed in the elimination sequence prior to this stage. This contradicts the fact that $u_i(s) \geq u_i(s^*)$. ∎

In particular, collusion-proofness holds for the coalition of all players. Hence, we get the following corollary:

Corollary 4.2 *The unique pure Nash equilibrium of a universally-max-solvable game is Pareto optimal.*

Private-information games. A game with private-information G is said to be universally-max-solvable if for every $u \in U$ the full-information game induced by players' actual utilities is universally-max-solvable.

Theorem 4.7 *If G is a universally-max-solvable game with private information, then best-reply dynamics are incentive-compatible in ex-post Nash (and even collusion-proof in ex-post Nash) for any best-reply mechanism that consists of at least $n(\Sigma_i m_i)$ steps, where n is the number of players, and m_i is the size of the action space of player i. The outcome of best-reply dynamics is the unique pure Nash equilibrium of the full-information game induced by G.*

Proof: We begin with a definition of a *phase* – a period of time in which every player has been activated and allowed to play at least once (In our current setting, as players are activated in round-robin order, a phase is a period of exactly n rounds. However, in asynchronous settings a phase can be a different period of time).

Let us examine the outcome that is reached if all players act according to a best reply strategy. Since the game is universally max-solvable, there exists a sequence of eliminations of universally-max-dominated actions that eventually leaves us with a single action profile. Let a_1 be the first action that is eliminated, and let i be the player to whom this action belongs. Player i is activated during the first phase, and changes his action to be the best-reply to the current action profile of the other players. Note, that no matter what the other players have chosen, the action a_1 is never the best-reply. By definition, there is always a different action (that may depend on what the other players are playing) that yields a higher payoff. Therefore, the strategy a_1 will no longer be played by player i after the first phase. From this point onwards, we are effectively playing in the game G_1 (in which strategy 1 has been eliminated). In this game, there is a universally-max-dominated strategy a_2 (that may belong to a different player) that is eliminated by the second phase for similar reasons.

More generally, in phase j, given that action $a_1, a_2, \ldots a_{j-1}$ are no longer played, the j'th action in the elimination order a_j will be considered by its player and will not be chosen as a best reply from that moment on (since other previous strategies are no longer played, no situation will ever arise in which a_j is the best-response). After $\sum_i (m_i - 1)$ phases, all of the actions in the elimination order are no longer played. The surviving action profile is then the Nash equilibrium of the original game, and no player deviates from it. This ends the game and sets this action profile as the outcome (and no penalty is activated). Since each phase lasts n rounds, we are assured of convergence within $(n-1)\sum_i m_i$ rounds.

Now we shall consider the case in which some group of players $I \subseteq [n]$ decides not to follow the best reply strategy. Since the mechanism gives out a penalty to the players if they did not converge after enough rounds of play, and because this penalty is strictly worse for any player than the payoff if gets if all follow best-reply, it cannot be that the group I gains from a situation in which there is no convergence. Therefore, we assume from this point on that the dynamics converged to an action profile $a = (a_1, a_2, \ldots, a_n)$. Specifically, the players in $[n] \setminus I$ are stable in the action profile a, and their action is already a best reply to the actions of the others. If a is the Nash equilibrium action profile, then the players in I have not gained from deviating. If it is not that action profile, then there are actions being played by some players in a that appear in the elimination sequence. let j be the player whose action is the earliest to be eliminated among all actions in a. Since no action before a_j in the elimination sequence is played by any of the players, a_j cannot be a best response to the current action profile. Therefore, player j is one of the members of the group I (all other players are stable). However, because a_j is universally-max-dominated, its best payment (after the elimination of all actions before it in the elimination sequence) is strictly lower than the payment of all other remaining strategies for player j, which specifically include the payment player j would get in the Nash equilibrium profile. Therefore player j loses from deviating from the best-reply strategy. This shows that any group of players that deviates must contain a player that loses by not playing best-reply, and so the best reply strategy profile is stable. ∎

This proof can easily be extended to prove analogous results for randomized best-reply mechanisms and for asynchronous best-reply mechanisms. Observe that the definition of universal-max-domination involved (for ease of exposition) strict inequalities. However, all of our definition easily carry over to the case of weak inequalities, as long as the prescribed best-reply strategies break ties in a way that is consistent with the order of an elimination sequence. In this case, there might be more than one pure Nash equilibrium, yet best-reply dynamics will still converge to a specific one.

4.4.2 Examples of Universally-Max-Solvable Games

1st-Price Auctions

Theorem 4.8 *1st-price auction games are universally-max-solvable.*

Proof: The 1st price auction game is a game with private information in which the actions of the players are their bids, and their utilities are a function of their private values (if a player wins $u_i = v_i - b_i$ otherwise $u_i = 0$). Observe, that leaving the item unsold is indeed a penalty. We prove that any full-information game induced by the actual utilities is universally-max-solvable by exhibiting an elimination sequence of universally-max-dominated action sets. We shall show how the first universally-max-dominated action-set T in the elimination sequence can be found. Finding the following universally-max-dominated action-sets is done via the recursive use of the same arguments.

Consider the case that there is a bidder i that has at least 2 actions, and the highest possible bid of this player is at least v_i. Then, we can choose T to be that player's highest bid (i's utility from bidding a bid higher or equal to v_i is at most 0). By repeating this argument, we are left with the case that all the possible actions of bidders (with more than one possible bid) are strictly less than their values for the item.

Let b_i be the lowest possible bid of player i. Let j be a bidder with at least 2 actions for which b_j is minimized (If more than one such bidder choose one that would lose if all of them bid their lowest bid. If there is no such bidder then there is a single action-profile and we are done). There are now two possibilities: Either there exists another bidder with at least 2 actions, or no such bidder exists. If there is such a bidder, then j never wins when bidding b_j, and therefore b_j is universally-max-dominated. Otherwise, each other bidder has only one possible bid. If j beats these bids when bidding b_j then b_j universally-max-dominates all other actions of j (and we can choose T to be the other actions of j). If this is not the case, then b_j is universally-max-dominated by j's other actions, and we choose T to be b_j.

It is easy to see that the tie-breaking rules implied by the the prescribed best-reply strategies are consistent with the elimination sequence described above. ∎

Congestion-Control Games

Theorem 4.9 *Congestion-control games are universally-max-solvable.*

Proof: A congestion-control game is a game with private information in which the actions of the players (flows) are their transmission rates, and their utilities are their throughput values (that cannot exceed their d_is). Observe, that not allowing flows to transmit is a penalty. We consider a full-information game induced by the actual utilities of the players, and show that it is universally-max-solvable. We do so by exhibiting an elimination sequence of universally-max-dominated action sets. We later discuss how to overcome the fact that the action sets are infinite.

Let e be an edge for which $\frac{c(e)}{k_e}$ is minimized, where k_e is the number of flows whose routes go through e. Consider the flow i with the lowest d_i whose route goes through e. If $d_i < \frac{c(e)}{k_e}$ then the set of transmission rates lower than d_i is universally-max-dominated by d_i (because of the way that capacity is shared by the routers). We can therefore remove all these actions. Observe, that the resulting subgame is equivalent to a congestion-control game in which i is removed and the capacity of all edges on r_i is decreased by d_i (for which we apply the arguments in this proof).

Otherwise, consider the set of transmission-rates of i lower than $\frac{c(e)}{k_e}$. Observe, that this set is universally-max-dominated by all actions greater or equal to $\frac{c(e)}{k_e}$ (again, because of the way that capacity is shared by the routers). This is true for all flows whose routes go through e. We can therefore remove all actions of these flows lower than $\frac{c(e)}{k_e}$.

Let e be defined as before. Observe, that in the subgame in which every node i whose route goes through e cannot choose an action lower than $\frac{c(e)}{k_e}$, all actions of such an i that are greater than $\frac{c(e)}{k_e}$ are universally-max-dominated by the action $\frac{c(e)}{k_e}$. Hence, we can iteratively remove universally-max-dominated action-sets until each flow whose route goes through e is only left with the action $\frac{c(e)}{k_e}$.

Observe, that in the resulting subgame every flow i whose route goes through e has to transmit at a rate of $\frac{c(e)}{k_e}$. This subgame is equivalent to the congestion-control game in which e is removed, all flows that go through e are removed, and for each flow whose route goes through e, we decrease the capacity of each edge on its route by $\frac{c(e)}{k_e}$. We apply the arguments in this proof to this new game.

It is easy to verify the the tie-breaking rules implied by the prescribed best-reply strategies are consistent with the described elimination sequence.

In order to see why the convergence rate is polynomial let us revisit the elimination sequence. Observe, that in every $2n$ rounds (going over all flows twice) we reach a subgame equivalent to a congestion-game in which either a flow or an edge is removed. It is easy to show, via similar techniques, that the resulting action-profile (that survives the elimination sequence) is optimal with respect to max-min fairness. ∎

Cost-Sharing Games

Theorem 4.10 *Cost-sharing games are universally-max-solvable.*

Proof: We demonstrate an elimination sequence in the game that leaves every player with only a single action profile. The first steps in our elimination sequence are to eliminate for each player the bids that are above his valuation v_i. These are clearly universally-max-dominated actions, as bidding above one's valuation can yield only one of two outcomes: Either the player is awarded the service and is then requested to pay the amount he bid (in which case he has a negative utility), or he is not awarded the service and does not pay (a utility of 0). For any other (lower) bid, his utility can be 0 or higher, and so the set of all other bids universally-max-dominates the set of bids above the valuation.

We now proceed to iteratively remove the bids of players. Let us assume that we are in step t of the elimination sequence. In any step, our current game G_t (after some actions have been eliminated) contains for each player a possible set of bids $B_{i,t}$ (initially this is the set $\{0, \dots, v_i\}$ since we've already eliminated higher bids). Let us define the set of players S_t as the set of players that have some chance of being awarded the service. I.e., let S_t be the set of players that have some remaining bid $b_i \in B_{i,t}$ that allows them to win in some eventuality (e.g., when any other player j bids the maximum: $\max(B_{j,t})$).

Throughout our elimination sequence the set S_t will be monotonically non-increasing. I.e., if at some point $i \notin S_t$ then $\forall t' > t \quad i \notin S_{t'}$. This property is maintained due to cross–monotonicity and the fact that we are only removing actions from the game. If actions are removed from some player's available set, he can bid no higher and is therefore not going to improve his ability to acquire the service. A player will also not gain from the removal of other players' strategies, since if they are suddenly not able to acquire the resource, the cost to the player itself may only increase or remain the same (due to the cross-monotonicity property). In any step t in our elimination one of the following cases must hold:

- **There exists a player with more than one action, and that player is not in S_t.** This player is never going to be awarded the service in G_t, and so will never pay his bid. We can therefore eliminate all bids except the highest one for such a player (since all actions yield a utility of 0, they are now universally-max-dominated in the weak sense by the highest bid that yields the exact same payoff). From now on, this player is effectively out of the game in one elimination step.

- **All players with more than one action are in S_t, but there exists a player in S_t for which $\min(B_{i,t}) < \rho(i, S_t)$.** Note that since players that are not in S_t have no chance of ever getting the service, this lowers the price that player $i \in S_t$ can ever hope to pay and be awarded the service to $\rho(i, S_t)$. Because of cross-monotonicity, a smaller group than S_t will increase the cost for player i, and no larger group is possible. Therefore, we can eliminate all bids below $\rho(i, S_t)$ for any such player. They never give a payoff of more than 0 in the game G_t and are therefore universally-max-dominated by higher bids that give a payoff of at least 0.

- **All players with more than one action are in S_t, and have $\min(B_{i,t}) \geq \rho(i, S_t)$.** In this case we can determine that all possible outcomes of the game G_t end with the set of player S_t being awarded the resource. This is because each of these players pays at least $\rho(i, S_t)$ – no other bids are available to them. Since this outcome is assured, each player is always better off paying as little as possible, and so we can deduce that all the higher bids of any players are universally-max-dominated by his lowest remaining bid. These eliminations bring us to a game in which all players have only a single strategy remaining and we are done.

Since we have shown that there are bids to be eliminated in any case (unless every player has only one strategy remaining, we have shown how to construct the full sequence of eliminations. ∎

Intern-Assignment Games

Theorem 4.11 *Intern-assignment games are universally-max-solvable.*

Proof:

We consider a simple condition on the preferences of the students in stable-roommates games, which we call “No Preference Cycle”.

Definition 4.8 *A Preference Cycle in the stable-roommates game, is a cyclic order defined over $k > 2$ students $p_1, \ldots, p_k$, such that each player p_i prefers p_{i+1} over p_{i-1} (All indices are considered modulo $k + 1$).*

No Preference Cycle holds if no Preference Cycles can be induced by the stable-roommates game. It is easy to show that in intern-assignment games there can be no Preference Cycles.

Claim 4.1 *No Preference-Cycle holds for any intern-assignment game.*

Proof: By contradiction. Assume that a Preference Cycle $p_1, ..., p_k$ exists. Consider a hospital p_i in the Preference Cycle (observe that a Preference Cycle always has the form ...-intern-hospital-intern-hospital-...). p_i must prefer p_{i+1} over p_{i-1}. Now, consider the hospital p_{i+2}. It prefers p_{i+3} over p_{i+1}. Hence, p_i also prefers p_{i+3} over p_{i+1}, which implies that it prefers p_{i+3} over p_{i-1}. If we continue this line of argument we can show that p_i's desire for an intern only increases as we move forward along the Preference Cycle. However, because of the cyclical structure of the Preference Cycle this would lead to a contradiction (because the intern that comes before p_{i-1} would be both preferred over p_{i-1} and less preferred than p_{i-1}. ∎

We shall prove the theorem for the more general case in which No Preference Cycle holds. A stable roommates game is a game with private information in which the actions of the players (students) are their choices of roommates. We define the utility utility u_i of player i as follows: Let v_i be a valuation function of i that assigns a value to every other student in a way that is consistent with i's preferences. For every set of actions (roommate-choices) $a = (a_1, ..., a_n)$, u_i has a value of $v_i(a_i)$ if i is preferred by a_i (given v_{a_i}) over all other students j for which $a_j = a_i$). Otherwise, $u_i(a) = 0$. That is, i get his value for the student he chose if he is making that student a better offer. Observe, that leaving all players unmatched is a penalty. By saying that the No Preference Cycle condition holds for a stable-roommates game we mean that it holds for any full-information game induced by the actual utilities of the players. We consider such an induced game and show that it is universally-max-solvable. We do so by exhibiting an elimination sequence of universally-max-dominated action sets. We shall show how the first universally-max-dominated action-set T in the elimination sequence can be found. Finding the following universally-max-dominated action-sets is done via the recursive use of the same arguments.

Pick a student i in the game and look at his most preferred roommate j. Then, look at j's most preferred roommate, and so on. Eventually one of the players must appear twice, thus forming a cycle. The cycle must be of length 2 because otherwise we get a Preference-Cycle. This implies two students who prefer each other over any other roommate. For each of these two students, the set of actions of selecting anyone other than the other student is universally-max-dominated by the action of selecting this ideal partner. We therefore eliminate all these actions (in two elimination steps), and remain with a subgame that is equivalent to a stable roommates game without these two students. Using the same line of reasoning we can find another pair of students that can be ideally matched in this subgame, and repeat the same procedure. ∎

Correlated Two-Sided Markets

Theorem 4.12 *Correlated two-sided markets are universally-max-solvable.*

Proof: We shall demonstrate an elimination sequence for the Correlated Two-Sided Market game. Let e be the edge in the graph that has the biggest weight. Both the buyer and the seller at each end of this edge prefer each other over all other players in the game. They are therefore able to make each other a better offer than any other player, and thus an action of choosing each other is guaranteed to give them their maximal payoff. It therefore universally-max-dominates all other actions they may chose (that yield a lower profit in any eventuality). So in two eliminations, the two players on e are left with only a single action and are effectively out of the game. The elimination sequence now proceeds as it would with the matching problem on the reduced graph (where the two players and all edges connected to them have been removed). ∎

Routing Games

Theorem 4.13 *Routing games are universally-max-solvable.*

Proof:

A routing game is a game with private information in which the actions of the players (vertices) are their choices of outgoing edges. A player's utility function u_i assigns a value to every action profile in a way that is consistent with the preferences. Observe, that not assigning outgoing edges to the vertices is a penalty. By saying that the No Dispute Wheel condition holds for a routing game we mean that it holds for any full-information game induced by the actual utilities of the players. (For a definition of Dispute Wheels see [48].) We consider such an underlying game and show that it is universally-max-solvable. We do so by exhibiting an elimination sequence of universally-max-dominated action sets. We shall show how the first universally-max-dominated action-set T in the elimination sequence can be found. Finding the following universally-max-dominated action-sets is done via the recursive use of the same arguments.

Fix a vertex a_0 with at least 2 actions. Let R_0 be a_0's most preferred existing route to d. Let a_1 be the vertex closest to d on R_0 such that u_1 prefers some other route R_1 to the suffix of R_0 that leads from a_1 to d.

Let us first handle the case that there is no such vertex a_1. This means that (u, d) is the most preferred route of the vertex u closest to d on R_0. Assume that u has at least two actions (otherwise, u always chooses (u, d) and hence we can apply the same arguments to the vertex that comes before u on R_0, and so on). This implies that (u, d) universally-max-dominates all other actions of u, which in turn implies that we can choose all other actions of u to be T.

If such an a_1 exists, then we choose a_2 to be the vertex closest to d on R_1 such that a_2's most preferred route R_2 is preferred over the suffix of R_1 that leads from a_2 to d. Once again if there is no such a_2 we are done (for the same reasons as before). Similarly, we choose $a_3, a_4, \ldots$. Since there is a finite number of vertices, at some point some vertex will appear twice in this sequence $(a_0, a_1, \ldots)$. This would result in the formation of a Dispute Wheel (in which the a_is are the pivot nodes and the R_is are the routes) – a contradiction. ∎

4.4.3 Beyond Universal-Max-Solvability

It was shown in [16] that for adword auctions with 3 slots (deterministic) best-reply dynamics are not even guaranteed to converge to a pure Nash equilibrium. [16] rectify this situation by introducing randomness into this setting. It can be shown that for the very restricted case of 2 slots and 2 bidders adword auctions are universally-max-solvable games (which implies, among other things, that deterministic best-reply dynamics always converge). However, we have strong suspicions that this is not true for the more general case. To prove that for adword auctions, in general, best-reply dynamics are still incentive-compatible, we must therefore resort to a completely different technique.

Definition 4.9 *Let M be a best-reply mechanism for which the prescribed best-reply dynamics always converge to a specific action-profile $x = (x_1, \ldots, x_n)$ (that is a pure Nash equilibrium of the induced full-information game). An action profile $a = (a_1, \ldots, a_n)$ is said to be bad (with respect to M) iff:*

- *There is a* unique *player i for which a_i is not a best-reply to the actions of the other players in a (for every other players j a_j is a best-reply to a).*
- *$u_i(a) > u_i(x)$ for some pure Nash equilibrium of the induced full information game.*

That is, an action-profile a is bad if only one player wishes to switch action, and that player is better off in that action-profile than it is in the action-profile to which best-reply dynamics converge.

Claim 4.2 *Let M be a best-reply mechanism for which the prescribed best-reply dynamics always converge to a specific action-profile $x = (x_1, \ldots, x_n)$. If no bad action-profile exists in a better-reply mechanism M then the prescribed best-reply dynamics are incentive-compatible in M.*

Proof: Assume, by contradiction that there is a player i that deviates from best-reply dynamics and gains by doing so. Let $a = (a_1, \ldots, a_n)$ be the action profile reached in the repeated-reply mechanism after i's deviation then it must be that $u_i(a) > u_i(x)$ (because i must prefer a to what it would have gotten had it not deviated). Observe that a must be such that i is the only player whose best-reply to a is not his action in a. This is because if there is another player j such that a_j is not a best-reply to j, then M would not have reached a (recall that a best-reply mechanism only reaches an action profile that isn't a penalty if all players "pass"). ∎

Claim 4.2 suggests a way for proving that best-reply dynamics is incentive-compatible in a repeated-reply mechanism M: Prove that the prescribed best-reply dynamics always converge to a specific action-profile in M. Then, prove that there are no bad action-profiles. This is precisely the approach we take in order to prove our result for adword actions. We shall refer to the randomized best-reply mechanism for adword auctions presented in Section 4.3 as the "adwords mechanism". [16] have shown that (randomized) best-reply dynamics converge to a unique pure Nash equilibrium (the one in which all bidders pay their VCG prices). So, it is up to us to show that there are no bad action profiles.

Theorem 4.14 *The two following properties hold for the adwords mechanism:*

- *The prescribed best-reply dynamics always converge to a specific action-profile. [16]*
- *There are no bad action-profiles.*

Remark 4.1 *We note that as the adwords mechanism is randomized, best-reply dynamics is guaranteed to converge to a specific action profile with high probability (arbitrarily close to* 1*). Therefore, as long as the adwords mechanism consists of sufficiently many steps, the possibility that the prescribed best-reply strategies do not converge has but a inconsequential affect on players' (expected) utilities. For ease of exposition we shall therefore ignore the effect of this event on players' (expected) utilities.*

Proof: It remains to show that there are no bad action profiles in the adwords game. Let us denote by $\pi(j)$ the player that ends up in the j'th slot if the Nash equilibrium is achieved (i.e., if all players follow the best-reply strategy and converge), and by b_i the bid submitted by player i in this scenario.

We shall assume the existence of a bad state s in which all players but one are stable, and will attempt to reach a contradiction. Let $\sigma(j)$ be the player that is allocated the j'th slot, and let b'_i be the bid of player i in this scenario.

If the unstable player does not win any slot, then he has a utility of 0, and is certainly not better off than in the Nash equilibrium outcome. We therefore assume that the unstable player occupies some slot in the state s. We begin by proving the following claim on the bids submitted by players:

Claim 4.3 *Let j be the slot that is occupied by the single non-stable player in the state s. In all lower slots, the valuation of the player that is awarded this slot is at least as high as the valuation of the player that got it in the Nash equilibrium. In addition, players in those slots bid at least as high as players in those slots do in the Nash equilibrium. That is:*

$$\forall j' > j \quad b_{\pi(j')} \leq b_{\sigma(j')} \quad \textit{and} \quad v_{\pi(j')} \leq v_{\sigma(j')}$$

if j is the lowest slot, the maximal bid of all losing players is also no lower than the maximal bid of losing players in the Nash equilibrium.

Proof: Notice that in the Nash equilibrium outcome, the losing players are the $n-k$ players with the lowest valuation, and that they all bid their valuation according to the best-reply strategy. In the state s, there are also $n-k$ players that are not awarded any slot, and they too bid their valuation (They must all be stable and thus follow the best-reply strategy). The highest bid they produce can therefore be no lower than the one given in the Nash equilibrium.

We now proceed to prove the claim by induction on the slot j'. If j is the lowest slot, then we are done. Otherwise, let $j' = k$ be the lowest slot. The amount payed by the player in this slot is determined by the highest bid of the losing players. This amount, as we have shown can only be greater than that of the Nash equilibrium. The player position k has a higher valuation than the losing players. This is because all losing players are bidding their valuation, and the player at position k he must have bid higher than them. We know that this bid is lower than his valuation (otherwise he would get a negative payoff and would not be stable – bidding lower would be preferable). Since there are at least $n-k$ players with lower valuations than player $\sigma(k)$, we are assured that his valuation is at least as large as that of player $\pi(k)$ that had *exactly* $n-k$ other players with a lower valuation than his own.

Since the bid of the highest losing player is no lower, and $v_{\sigma(k)} \geq v_{\pi(k)}$ we can deduce that $b'_{\sigma(k)} \geq b_{\pi(k)}$. This is because the player at the k'th slot pays his price of indifference for the next slot which is defined as:

$$b_{\pi(k)} = \frac{\alpha_k}{\alpha_{k-1}} \cdot b_{\pi(k+1)} + \left(1 - \frac{\alpha_k}{\alpha_{k-1}}\right) \cdot v_{\pi(k)}$$

and we have:

$$b_{\pi(k)} = \frac{\alpha_k}{\alpha_{k-1}} \cdot b_{\pi(k+1)} + \left(1 - \frac{\alpha_k}{\alpha_{k-1}}\right) \cdot v_{\pi(k)} \leq \frac{\alpha_k}{\alpha_{k-1}} \cdot b'_{\sigma(k+1)} + \left(1 - \frac{\alpha_k}{\alpha_{k-1}}\right) \cdot v_{\sigma(k)} = b'_{\sigma(k)}$$

This concludes the proof for the base of the induction.

In the induction step we shall assume that the claim has been proven up to slot $j' + 1$ and will prove it for slot j'. The player in position j' must have a higher value than the player in position $j' + 1$. This is because the player in position $j' + 1$ is stable, and is bidding its price of indifference – the highest price it will be willing to pay to get slot j' rather than $j' + 1$. The player at position j' is indeed paying this price and is stable. That is, it is gaining more than it would gain for slot $j' + 1$. This implies that his price of indifference for slot $j' + 1$ would have been higher, and so it must be that $v_{\sigma(j')} > v_{\sigma(j'+1)}$. From the induction assumption we have that $v_{\sigma(j'+1)} \geq v_{\pi(j'+1)}$ this then implies that $v_{\sigma(j')} > v_{\pi(j'+1)}$, but $v_{\pi(j')}$ is the next valuation after $v_{\pi(j'+1)}$ and no player has valuation between them (we know this because in the Nash equilibrium outcome players are awarded slots in an order that is consistent with their valuations). Therefore, it must be that $v_{\sigma(j')} \geq v_{\pi(j')}$ as needed.

Now, since in both states the player in position j' is stable, it is bidding its price of indifference.

$$b_{\pi(j')} = \frac{\alpha_{j'}}{\alpha_{j'-1}} \cdot b_{\pi(j'+1)} + \left(1 - \frac{\alpha_{j'}}{\alpha_{j'-1}}\right) \cdot v_{\pi(j')} \leq \frac{\alpha_{j'}}{\alpha_{j'-1}} \cdot b'_{\sigma(j'+1)} + \left(1 - \frac{\alpha_{j'}}{\alpha_{j'-1}}\right) \cdot v_{\sigma(j')} = b'_{\sigma(j')}$$

■

Using our claim we can deduce that the unstable player $\sigma(j)$ is forced to pay for the j'th slot a price that is at least as high as the price it cost in the Nash equilibrium outcome (since it pays the bid of the player below it which could only increase). However, in the Nash outcome he could have bid for that position and gotten this payment but did not (Note that there is a case in which positions j and $j + 1$ are awarded to two players with the same bid and the player $\sigma(j)$ cannot get position j because ties are not broken in the right way. In this case, if he goes for position $j - 1$ he will pay the same amount and will only gain more clicks for the same price). This brings us to a contradiction. It cannot be that the unstable player is better off than in the Nash equilibrium. ■

Chapter 5

Incentive-Compatibility Can Lead to Bad Approximations

5.1 Introduction

In the previous chapters, we have presented several realistic settings in which one can achieve different design requirements. One important such requirement is that of incentive-compatibility. In this chapter, and the subsequent one, we shall study the obstacles that face us when we attempt to design incentive-compatible algorithms (protocols). Both this chapter and the next one focus on the complex interplay between incentive-compatibility *and computational-efficiency.*

In this chapter we establish the first significant approximability gap between algorithms that are both truthful and computationally-efficient, and algorithms that only achieve one of these two desiderata. Our result is actually made up of two complementary results – one in the communication-complexity model and one in the computational-complexity model. Our computational-complexity result is one of the first impossibility results connecting mechanism design to complexity theory; its novel proof technique involves an application of the Sauer-Shelah Lemma and may be of wider applicability.

5.1.1 Combinatorial Public Projects

In real 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.

This is a typical instance of a general problem we define, called COMBINATORIAL PUBLIC PROJECT PROBLEM, or CPPP. 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 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 Subsec 5.1.2). The objective is to find a subset of the resources S of size k, where k is a parameter, which maximizes the *social welfare*, i.e., the sum of agents' values for the chosen subset $\Sigma_i \ v_i(S)$.

Computationally speaking, this problem is relatively benign; while many of its special cases are NP-hard [30] it is well known [73] that the greedy algorithm achieves a constant approximation ratio (specifically, $1 - \frac{1}{e}$).

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[1], and this misrepresentation makes optimization impossible. There is a very general method for providing incentives for the agents to reveal their true valuation, namely the *Vickrey-Clarke-Groves (VCG) mechanism* [100, 17, 50]. However, VCG requires that we solve *exactly* (typically many instances of) the CPPP — 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 [75].

In other words, efficient approximability and incentive compatibility seem to be at loggerheads. This tension underlies much of the work in algorithmic mechanism design [75, 76, 62, 21]. In this chapter, we establish a huge gap between the quality of the solutions that can be obtained by algorithms for CPPP 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 CPPP can obtain any reasonable approximation ratio, specifically, a ratio better than $\sqrt{m}$ (this holds even for the case of two agents)[2]. Our inapproximability result settles a long-standing open question in algorithmic mechanism design [40, 86]: we exhibit a problem (CPPP) 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 (does not allow for constant approximations) if we care about both[3].

Our result is actually made up of two complementary results – one in the communication-complexity model and one in the computational-complexity model. Technically speaking, each of these results must overcome two main challenges [86]: First, we must provide a combinatorial characterization of truthful algorithms. Then, once we have such a characterization, we can exploit it to prove an inapproximability result.

An interesting way to view our results 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

[1] As we are aiming for impossibility results we restrict our attention to the centralized setting in which some trusted computational centre is communicating with the selfish agents, and computes the outcome based on the information they provide. Unlike distributed settings (like interdomain routing) in which the agents take part in the computation of the outcome, and can therefore manipulate an algorithm in many different ways, here the only form of manipulation available to an agent is "lying", i.e., misreporting his preferences. Incentive-compatibility therefore boils down to truthfulness. 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. In most of the works in the field of algorithmic mechanism design the very robust notion of equilibrium in dominant strategies is used. It is well known (by the Revelation Principle) that, without loss of generality, we can limit ourselves to only considering "incentive compatible" mechanisms, also known as "truthful" mechanisms or "strategy-proof" mechanisms. In such mechanisms participants are always rationally motivated to correctly report their private information.

[2] We note that a simple truthful polynomial-time algorithm that obtains a $\sqrt{m}$ approximation ratio for CPPP shows that our result is tight in both models. [92]

[3] It is known that algorithms that are both truthful *and* computationally-efficient might not perform (in terms of approximation ratio) as well as algorithms that only meet *one* of these two requirements. However, it was unclear by how much their performance is harmed. In fact, so far researchers have only been able to establish a gap of 2 between these two types of algorithms [62, 24].

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,[4] classical VCG theory implies that P, NP, and NP-hardness are preserved under truthfulness. *Our computational-complexity result essentially states that APX is* not *preserved* (our communication-complexity result proves an analogous statement in the communication model). We discuss this novel approach to algorithmic mechanism design in Section 5.4.

Communication-complexity lower bound. Our first main contribution is an exponential lower bound on the amount of communication required by *any* truthful algorithm that approximates the CPPP within any reasonable ratio:

Theorem: *Any truthful algorithm for the* CPPP *that obtains an approximation ratio of $O(m^{\frac{1}{2}-\epsilon})$ requires communication that is exponential in m (for every $\epsilon > 0$ and even for $n = 2$).*

The first step of the proof is showing that any truthful algorithm for CPPP must be an affine maximizer. An affine maximizer is an algorithm defined by a fixed set of solutions, and which, given a set of valuations, picks the solution in the set that maximizes social welfare. A celebrated result by Roberts [84] essentially states that if an algorithm is truthful for all valuations, then it has to be an affine maximizer. However, this result does not necessarily hold when restricted to special settings (submodular valuations, etc.). In fact, for many restricted settings, truthful algorithms that are *not* affine maximizers are known (e.g. [8, 5]). In the present case, by carefully applying machinery from Roberts's (long and delicate) proof — actually, its recent interpretation by Lavi, Mu'alem and Nisan [62, 63] — we are able to *prove that any truthful algorithm for the nondecreasing submodular case of the* CPPP *is an affine maximizer.* The proof explores the geometric and topological properties that must hold for any truthful algorithm for CPPP to prove affine maximization.

After providing this combinatorial characterization of truthfulness the second step is proving a lower bound for affine maximizers. 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 our communication-complexity lower bound. The proof of the theorem draws ideas from works in many different fields of research (e.g., [84, 80, 74, 62, 63, 21, 24, 32, 67]), and so we defer a discussion of the related work to Section 5.2.

Computational-complexity lower bound. Our previous result is a *communication-complexity* lower bound. Informally, the agents are assumed to compute their valuations based on exponentially large data, and then it is shown that too much of this data must be exchanged for the algorithm to be truthful. However, many interesting valuations depend on very succinct data (recall the introductory overlay problem, in which the input is a distance matrix), and *computational-complexity* techniques would be needed to establish lower bounds for these; *no such results had been known* (with the possible exception of Theorem 5 in [62]). Our next major result does exactly this: We consider a class of nonnegative submodular (though not nondecreasing) valuations, that are succinctly described. We show that, despite the fact that a $1 - \frac{1}{e}$ approximation exists for this class, no truthful and polynomial-time algorithm can obtain an approximation ratio (asymptotically) better than $\sqrt{m}$.

As in our communication-complexity result, the proof of this result consists of two parts: Proving that any truthful algorithm must be an affine-maximizer, and proving a hardness result for affine-maximizers. The main challenge now lies in the second part (the first part is basically derived from the characterization of truthfulness in our communication complexity result). The heart of our

[4] See discussion about other optimization goals in [72].

proof is therefore showing the following inapproximability result for a class of succinct submodular functions we call "*coverage-penalty valuations*":

Theorem: *For* CPPP *with coverage-penalty valuations there is no polynomial-time affine maximizer with $O(m^{\frac{1}{2}-\epsilon})$ approximation ratio unless $NP \subseteq BPP$ (for every $\epsilon > 0$).*

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 [90, 94][5] to obtain a hardness result. This result has interesting projections on complexity theory as discussed in section 5.3.

Organization of the chapter. In the balance of this section we introduce the CPPP and the communication- and computational-complexity notions. In the next section we prove our exponential communication-complexity lower bound, while in Section 5.3 we prove our computational-complexity lower bound. In Section 5.4 we discuss the implications of our results in terms of the traditional computational-complexity classes. In Section 5.5 we present several open questions.

5.1.2 Definitions

In CPPP there is a set of n *agents* $\{1, ..., n\}$, and a set of m *resources* $\{1, ..., m\}$. Each agent has a *private valuation function* $v_i : 2^{[m]} \rightarrow \Re_{\geq 0}$. 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 $v_i(\emptyset) = 0$. We assume that every v_i is *submodular*, i.e., for every $S, T \subseteq [m]$ $v_i(S \cup T) + v_i(S \cap T) \leq v_i(S) + v_i(T)$. 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 $j \notin T$ $v_i(S \cup \{j\}) - v_i(S) \geq v_i(T \cup \{j\}) - v_i(T)$. Submodularity arises in many contexts – both economic and computational (see [22, 23, 31, 33, 32, 64, 101] and references therein). This chapter focuses on submodular functions that are *nondecreasing*[6] in that $S \subseteq T$ implies $v(S) \leq v(T)$. Nondecreasing submodular functions are known to have particularly good properties; for example, they can be approximated within a ratio of $1 - \frac{1}{e}$ (for general nonnegative submodular functions a constant approximation ratio exists [32]).

The objective in the CPPP 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 $T \in argmax_{S \subseteq [m],\ |S|=k}\ \Sigma_i v_i(S)$.

We are interested in algorithms (*mechanisms*) for CPPP satisfying three desiderata:

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.

Computational efficiency. Our algorithms should run in time that is polynomial in the natural parameters of the problem – m and n. However, as the "input" (the data enabling each agent to compute the valuation) can be exponential in m we must specify how it can be accessed. In mechanism design one often takes a "black box" approach (see [22]): 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 the *value query model* a query is a subset of resources $S \subseteq [m]$, and the answer is simply $v_i(S)$; This weaker model is mainly used for designing algorithms.

[5] This lemma is closely related to the notion of the Vapnik-Chervonenkis (VC) dimension.

[6] We slightly relax this assumption in Section 5.3.

- The *general query* model is equivalent to *Yao's communication model* [102, 60], 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 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 m and n. We follow this approach in Section 5.3.

Truthfulness. 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 utility (the value of the set chosen by the algorithm minus the payment assigned to him) by misreporting his valuation[7]. Formally, for every $i \in [n]$ we have that[8]:

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

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.

5.2 Communication-Complexity of Mechanism Design

In this section we prove the following:

Theorem 5.1 *Any truthful algorithm for* CPPP *that obtains an approximation ratio of* $O(m^{\frac{1}{2}-\epsilon})$ *requires exponential communication (for every* $\epsilon > 0$ *and even for* $n = 2$*).*

The proof proceeds in two steps: We first establish (Lemma 5.1) that any truthful algorithm for the CPPP must be of a very restricted kind called an *affine maximizer.* We then show that any affine maximizer for the CPPP requires exponential communication (Lemmas 5.2 and 5.3). This second part uses techniques introduced by Dobzinski and Nisan [21] for proving communication complexity lower bounds for affine maximizers.

The Characterization Lemma is the heart of our proof. It establishes that the CPPP has a rich enough structure, and appropriately strong interaction between the agents, so that a subtle variant of Roberts's proof [84] is enabled. The line of argument used in our proof follows that of Roberts', as presented by Lavi, Mu'alem, and Nisan [63]. However, applying this machinery to our setting is not straightforward as we are dealing with a *restricted* domain of valuation functions (submodular, nondecreasing). Therefore, we must handle various technical difficulties: We must ensure that the valuation functions we construct belong to this restricted domain. We must also take measures to deal with the fact that our valuation domain is *not open* (in the standard topological sense). Unlike Roberts' proof, our proof makes repetitive use of the *strong monotonicity* constraint introduced in [62] (also see [24]).

[7] 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.

[8] In this chapter we are only dealing with deterministic algorithms, and define truthfulness accordingly.

5.2.1 The Characterization Lemma

Let A be an algorithm for the nondecreasing submodular CPPP with m resources and parameter k. We define A's *range* R_A to be the resource subsets of size k that are output by A for *some* input, i.e., $R_A = \{S| \; \exists v = (v_1, ..., v_n) \; s.t. \; A(v) = S\}$. Informally, A is an affine-maximizer if it *always* optimizes over its entire range R_A.

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

$$\forall v = (v_1, ..., v_n)$$

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

Lemma 5.1 *Any truthful algorithm for the nondecreasing submodular* CPPP *with $n = 2$ is an affine-maximizer.*

Remark 5.1 *We note that, as we are aiming for an impossibility result, we prove the characterization lemma only for the 2-agent case. We shall later show that our inapproximability result holds even for this special case.*

Proof: As is common in such proofs (see for example [84, 63]), we study the topological structure of vectors of *valuation differences.* For any pair of valuation functions $v = (v_1, v_2)$, we shall denote by $v(S) - v(T)$ the vector in $\Re^2$ $(v_1(S) - v_1(T), v_2(S) - v_2(T))$. We also define

$$P(S,T) = \{\alpha \in \Re^2 \; |\exists \; v \; s.t. \; A(v) = S \; and$$

$$v(S) - v(T) = \alpha\}.$$

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 $\{C_R\}_{R \in R_A}$)

$$(\Sigma_i w_i v_i(S)) + C_S \geq (\Sigma_i w_i v_i(T)) + C_T,$$

which implies that

$$\Sigma_i w_i (v_i(S) - v_i(T)) + (C_S - C_T) \geq 0.$$

Informally, observe that the inequalities above suggest that if A is an affine maximizer, then there is a line l in $\Re^2$ 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 [63] for an explanation of the geometric intuition behind the proof of Roberts' Theorem (which also underlies our proof).

Strong monotonicity. We shall require the strong monotonicity property.

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

Proposition 5.1 *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 (for some arbitrarily large $\beta > 0$):

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

It is easy to verify that this is indeed a nondecreasing submodular function (this construction is inspired by [32, 67]). 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 proof.

Claim 5.1 *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 5.1) and of Proposition 5.1 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 5.1 tells us something important about the structure of the different $P(S,T)$ sets in $\Re^2$: 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 nonincreasing slope. However, we do not yet know that this nonincreasing 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 propositions:

Proposition 5.2 *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 5.1 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 5.1 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 5.3 *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 nonnegative 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 5.1 (as in the proof of Proposition 5.2) 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 $P(S,W)$ $\alpha + \alpha' \in P(S,W)$. ∎

We shall 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 proof for the more general case is achievable via the exact same logic as presented in [63] (claim 4 and the following claims in the first proof in that paper follow from what we have proven thus far).

Corollary 5.1 *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 nonincreasing) must be a straight line[9]. 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.

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 $v' \in V$.

Proposition 5.4 *X is convex.*

Proof: We first show that if $\alpha, \alpha' \in X$ then $\frac{\alpha+\alpha'}{2} \in X$. Suppose, by contradiction, that $\alpha, \alpha' \in X$ but $\frac{\alpha+\alpha'}{2} \notin X$. By Proposition 5.3 $\alpha + \alpha' \in X$, and by Proposition 5.2 $-\frac{\alpha+\alpha'}{2} \in X$. However, Proposition 5.3 now implies 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 (and the openness of V) 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 of the axes $\overrightarrow{0}$, as by Proposition 5.2 $\alpha \in X$ 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 nonincreasing). 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 lemma.

∎

Removing the Strong Monotonicity and Open Domain Assumptions. So far, the proof of the Characterization Lemma relied on the assumptions that the algorithms in question are strongly monotone and that the domain of valuations is open. Here we exploit the following machinery to do away with these:

[9] 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 $\Re^2$, and their interiors are disjoint (by Proposition 5.2), then they must be separated by a straight line.

Theorem 5.2 *[62] 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 5.2 implies the following modus operandi: We shall focus on a "rich" open subdomain of the domain of all nondecreasing 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 5.2) this will conclude the proof. We now provide a way to slightly tweak all constructions of valuation functions in Subsections 5.2.1 and 5.2.2 to ensure that they all belong to this open domain.

Specifically, we define the *strict domain* V^{strict} to be the domain of all valuations that are *strictly* nondecreasing (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., $\forall S \subset T \subseteq [m]$, $\forall j \notin T$ and $\forall i \in [n]$ $v_i(S \cup \{j\}) - v_i(S) > v_i(T \cup \{j\}) - v_i(T)$).

Claim 5.2 *V^{strict} is an open domain.*

Proof: 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 show that any nondecreasing and submodular valuation function can be slightly tweaked to fit in V^{strict}. Hence, we can prove our theorem for the case that all valuations ar slightly tweaked in this manner. The reader may verify that (for sufficiently small choices ϵ) all the arguments in this section also apply to these slightly different constructions of valuation functions.

Claim 5.3 *For any nondecreasing and submodular valuation function $v_i \in V_i$, and any $\epsilon > 0$, there exists a valuation function $v'_i \in V_i$ that is strictly nondecreasing and has strictly decreasing marginal utilities such that $\forall S \subseteq [m]$ $|v_i(S) - v'_i(S)| \leq \epsilon$.*

Proof: Fix some $\epsilon > 0$ and valuation function v_i that is nondecreasing and submodular. Let $\epsilon' << \epsilon$. 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 nondecreasing and has strictly decreasing marginal utilities. Moreover, for any nondecreasing submodular valuation function v and for any ϵ' it holds that $v'_i = v_i + g_{\epsilon'}$ is *strictly* nondecreasing and has *strictly* decreasing marginal utilities. ∎

5.2.2 Lower Bound for Affine Maximizers

Let k, the number of chosen resources in the CPPP, be fixed to $\sqrt{m}$ throughout this subsection. We shall now show that any algorithm A that obtains a reasonable approximation ratio must have a range R_A of exponential size. We will then exploit affine maximization to show that the size of the range is a lower bound on the communication complexity.

Lemma 5.2 *For any algorithm A that obtains an approximation ratio of $m^{\frac{1}{2}-\epsilon}$ to the optimal social welfare it must hold that $|R_A| = \Omega(e^{m^\epsilon})$, even for $n = 1$.*

Proof: (Sketch) Consider an algorithm A with range R_A. Choose a set of resources $V \subseteq [m]$ by sampling $[m]$ independently with probability $p = m^{-\frac{1}{2}+\epsilon}$, where $\epsilon > 0$ is small and fixed, and consider the valuation function $v(S) = |S \cap V|$. We claim that, unless R_A contains at least $\Omega(e^{m^\epsilon})$ sets, with high probability it approximates v very poorly.

Indeed, it follows from the Chernoff bound that, for any small $\delta > 0$, with probability at least $1 - |R_A| \cdot \exp(\delta^2 m^{\frac{1}{2}+\epsilon})$, in this instance of CPPP the optimum $|V|$ is at least $(1-\delta)m^{\frac{1}{2}+\epsilon}$, while the solution returned by the affine maximizer will be at most $(1+\delta)m^\epsilon)$. The lemma follows. ∎

The following now concludes the proof of our main result:

Lemma 5.3 *Any affine maximizer A with range R_A requires $\Omega(|R_A|)$ communication, even for $n = 2$.*

Proof: (Sketch) 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. 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: $v_i(T) = \beta \cdot (\Pi_{S \in R_A^i} |S| - \Pi_{S \in R_A^i}(|S| - |S \cap T|))$ $\forall T \subseteq [m]$. 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 communication set-disjointness problem, establishing that $\Omega(|R_A|)$ communication is required. ∎

5.3 Computational-Complexity of Mechanism Design

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

Theorem 5.3 *There is a class of succinctly-described, nonnegative and submodular valuation functions C such that:*

- *There is a polynomial-time algorithm for the* CPPP *with valuations in C that approximates the optimal social-welfare within a ratio of* $1 - \frac{1}{e}$.
- *Any truthful and polynomial-time algorithm cannot achieve an approximation ratio better than $m^{\frac{1}{2}-\epsilon}$ unless NP $\subseteq$ BPP (for every $\epsilon > 0$).*

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.

5.3.1 Lower Bound For Affine Maximizers.

We define a class of succinctly described valuation functions called *"coverage-penalty valuations"*. A coverage-penalty valuation $v_{F,T}$ is defined by a family $F = R_1, ...R_m$ of m subsets of a universe U, and a subset $T \subseteq [m]$: $\forall S \subseteq [m]$, $v_{F,T}(S) = |\bigcup_{j \in S} R_j| - \frac{\epsilon}{|T|}(\max\{0, |S \cap T| - \frac{|T|}{2}\})$ (for some extremely small value of ϵ). Intuitively, for every $S \subseteq [m]$, $v_{F,T}$ assigns a value that equals the number of elements in U covered by the union of the sets in F with indices in S, minus an insignificant "penalty" if S has "too many" elements in common with T. It is easy to verify that this function is indeed nonnegative[10] and submodular (but not nondecreasing).

Observe that for any value of k, it is easy to obtain a $1 - \frac{1}{e}$ approximation ratio for CPPP with coverage-penalty valuations (simply ignore the "penalty terms" in the definitions of the v_i's and apply the $1 - \frac{1}{e}$ greedy approximation algorithm). We shall now show that despite this fact, any affine-maximizer fails to obtain a reasonable approximation ratio:

Theorem 5.4 *No polynomial-time affine maximizer for* CPPP *with coverage-penalty 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: (Sketch) 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 5.2, it can be shown that R_A must contain $\Omega(e^{m^\epsilon})$ sets. We now recall the Sauer-Shelah Lemma:

Lemma 5.4 *([90, 94]) 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 problem in which we have a family F' of t subsets of a universe U and wish to determine whether there are $\frac{t}{2}$ sets in F' whose union covers the entire universe (which is known to be NP-complete). The embedding is as follows: We construct a coverage-penalty valuation function v for which each element in M' corresponds to one of the t subsets in F', and all other elements in $[m]$ correspond to empty sets. We set the "penalty set" T to equal M'. Now, observe that the value of the set output by the affine maximizer A equals $|U|$ *iff* there are $\frac{t}{2}$ sets in F' whose union covers U.

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 [83, 91, 69] 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 [2] we can turn the reduction described above into a *probabilistic* polynomial-time reduction (thus concluding the proof of the theorem).

Lemma 5.5 *([2]) 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 $j \in [q]$.*

[10] This may not be true if some of the R_i's are the empty set. However, this is easily handled by simply adding ϵ to $v_{F,T}(S)$ for every S.

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 5.5, and then associating each subset of the universe in the NP-complete problem with *all* elements in one of the pairwise-disjoint subsets. ∎

5.3.2 Connection to Complexity Theory

This technique 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 $\textsc{sat}_L$ to be the problem: "Given a CNF, is there a truth assignment *in* L that satisfies it?" The proof technique implies that:

Theorem 5.5 *Let L be any exponentially dense language. If $\textsc{sat}_L$ is in P, then NP $\subseteq$ P/poly.*

Observe that 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 5.6 *Let L be any exponentially dense language. If CIRCUIT $\textsc{sat}_L$ 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. ∎

It would be very interesting to extend this idea (or the ideas in the proof of Lemma 5.5) to $\textsc{sat}_L$.

5.3.3 Extension 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 (Lemma 5.1 in Section 5.2) 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 Lemma 5.1 *and* all coverage-penalty valuations.

It is easy to see that an approximation of $1 - \frac{1}{e}$ is achievable for the CPPP when the valuation functions are in C (this is because coverage-penalty valuations are arbitrarily close to nondecreasing submodular valuations and all other valuations in C are nondecreasing and submodular). Because of the way we defined C one can show (by carefully applying the proof of Lemma 5.1 to this case) that any truthful algorithm for the CPPP with valuations in C is an affine-maximizer. We can now use Theorem 5.4 to obtain our result.

5.4 Is APX Closed Under Truthfulness?

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. What becomes of these traditional complexity classes in this new regime? That is, how much harder do problems in these complexity classes become if we insist on the truthfulness requirement? To fully answer these questions we must elaborate some more on the hardness of achieving truthfulness in various settings.

5.4.1 Mechanism Design Environments

The setting. Consider the following general mechanism design setting: There is a finite set of *outcomes* $O = \{a, b, c, ...\}$, and a set of *strategic agents* $N = \{1, ..., n\}$. Each agent i has a *valuation function* $v_i : A \rightarrow R$, taken from *valuation space* V_i. that is his private information. The agents are self-interested and only wish to maximize their own gain. The global goal is expressed by a *social choice function* f that assigns every possible n-tuple of agents' valuations $(v_1, ..., v_n)$ an outcome $a \in O$. We say that an algorithm (*mechanism*) computes f if its output for any profile of valuations is that of f.

We want an algorithm (mechanism) to be such that the agents are rationally motivated to execute it. This can be achieved via a *payment function* p which, for every n-tuple of valuation functions, demands a payment from each agent. p is such that no agent can increase his utility (the value of the outcome the algorithm outputs minus the payment assigned to him) by not adhering to the algorithm. A mechanism M is said to *truthfully implement* (or simply implement) a social choice function if its outcome for every n-tuple of agents' valuations matches that of the social-choice function, and if it enforces payments of the different agents in a way that motivates truthful behavior. Formally, for every $i \in [n]$ we require that:

$$\forall v_i, v'_i \in V_i, \forall v_{-i} \in V_{-i},$$

$$v_i(M(v_i, v_{-i})) - p(v_i, v_{-i}) \geq v_i(M(v'_i, v_{-i})) - p(v'_i, v_{-i}),$$

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 $M(v)$ is the set M outputs for the valuation profile v.

Utilitarian and non-utilitarian social-choice functions. A very common social choice function is the *utilitarian* function; A utilitarian function always maximizes the *social welfare*, i.e., finds the outcome a for which the expression $\Sigma_i v_i(a)$ is maximized. Utilitarian functions represent the "overall content" of the agents, as they maximize the sum of agents' values. This notion of *fairness* is but one of several that have been considered (explicitly and implicitly) in mathematical, economic and computational literature.

A well known example of non-utilitarian fairness is the cake-cutting problem, presented by the Polish school of mathematicians in the 1950's (Steinhaus, Banach, Knaster). In problems like the cake-cutting problem the notion of fairness we strive for is *max-min fairness*; for every n-tuple of v_i valuations the max-min function assigns the outcome a that maximizes the expression $min_i\ v_i(a)$.

Intuitively, the max-min function chooses the outcome $a \in A$ in which maximizes the satisfaction of the least satisfied agent.

While in many computational and economic settings the social choice function we wish to implement in a truthful manner is utilitarian (e.g., CPPP, social-welfare maximization in combinatorial auctions [12]), often this is not the case. Problems in which the social choice function is non-utilitarian include revenue maximization in auctions (e.g. [44]), minimizing the makespan in scheduling (e.g. [76, 6]), minimizing congestion in communication networks, etc.

A classic result of mechanism design states that for every utilitarian problem there exists a mechanism that truthfully implements it – namely, a member of the celebrated family of *VCG mechanisms* [100, 17, 50]. No general technique is known for truthfully implementing non-utilitarian social-choice functions. In fact, some non-utilitarian social-choice functions cannot be truthfully implemented [62, 76]. Hence, from a computational point of view it is natural to ask how well these social choice functions can be *approximated* in a truthful manner. This can be seen as the mechanism-design analogue of the well known price of anarchy problem [59, 87]. In the next subsection we show that sometimes the truthfulness requirement alone (even without computational limitations) is sufficient to make optimization, or even approximation, impossible.

5.4.2 Sometimes Truthfulness Alone Can Be Hard

Consider the following simple auction setting: Two items a, b are to be sold to two bidders $1, 2$. Each bidder i has a valuation function v_i that assigns a value for item a ($v_i(a)$) and a value for b ($v_i(b)$). The goal is to allocate one item to each bidder in a way that maximizes the the max-min fairness. We prove the following impossibility result:

Theorem 5.7 *No truthful mechanism can obtain* any *approximation ratio to the optimal max-min fairness value in this auction.*

Proof:

We start by defining a useful property of mechanisms called "*weak monotonicity*":

Definition 5.1 *Let M be a truthful mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v'_i be a valuation function. Denote by a the outcome M outputs for v and by b the outcome that M outputs for (v'_i, v_{-i}). M is said to be weakly monotone if for all such i, v, and v'_i it holds that: $v_i(a) + v'_i(b) \geq v'_i(a) + v_i(b)$.*

[9] proves that any truthful mechanism must be weakly-monotone. For completeness, we present this simple proof.

Lemma 5.6 *Any truthful mechanism must be weakly monotone.*

Proof: Let M be a truthful mechanism. Let $i \in [n]$ and let $v = (v_1, ..., v_n)$ be an n-tuple of agents' valuations. Let v'_i be a valuation function. Denote by a the outcome M outputs for v and by b the outcome that M outputs for (v'_i, v_{-i}). Consider agent i. It is well known that the price an agent is charged by the mechanism to ensure his truthfulness cannot depend on the agent himself. Hence, the payment of agent i in a and b is a function of v_{-i} and of a and b respectively. We denote by $p_i(v_{-i}, a)$ and by $p_i(v_{-i}, b)$ i's payment in a and b respectively. It must hold that $v_i(a) - p_i(v_{-i}, a) \geq v_i(b) - p_i(v_{-i}, b)$ (for otherwise, if i's valuation function is v_i, he would have an incentive to declare his valuation to be v'_i). Similarly, $v'_i(b) - p_i(v_{-i}, b) \geq v'_i(a) - p_i(v_{-i}, a)$. By summing over these two inequalities we get the weak monotonicity inequality. ■

Going back to our auction setting, let $c > 0$. We define the following valuation functions: $v_1(a) = 2 \;\; v_1(b) = \frac{1}{c} \;\; v_2(a) = 4 - \epsilon \;\; v_2(b) = 1 + \epsilon$. Note, that the optimal allocation assigns a to agent 1 and b to agent 2, thus obtaining a max-min value of $1 + \epsilon$. Also note, that this allocation will also be chosen by any c-approximation mechanism.

We alter the valuation of agent 2 into v'_2 such that $v'_2(a) = \frac{1}{c} \;\; v'_2(b) = \frac{1}{c^2} - \epsilon$. The optimal max-min value is now $\frac{1}{c}$. Observe that any c-approximation mechanism must assign item b to agent 1 and item a to player 2. However, if this happens we have that:

$$(1 + \epsilon) + \frac{1}{c} = v_2(b) + v'_2(a) < v_2(a) + v'_2(b) = (4 - \epsilon) + \frac{1}{c^2}$$

This violates weak monotonicity, and so no truthful c-approximation mechanism exists. Since this is true for any $c > 0$ the theorem follows. ■

5.4.3 Complexity Classes and Truthfulness

To explain the intricate connection between traditional complexity classes and truthfulness we shall require the following definitions:

Definition 5.2 *We say that an optimization problem is decentralized if the input is assumed to be provided by different players, and if the part of the input held by each player i is a function v_i, representing that player's preferences over the possible outcomes, that assigns a real numerical value to each such outcome.*

Definition 5.3 *The mechanism design version L_{MD} of a decentralized optimization problem L, is the optimization problem that is identical to L, with the exception that for every input the objective now also consists of assigning payments that induce truthful behaviour by the players (as explained in this section).*

Observe that since sometimes no truth-inducing payments exist for a certain problem, it may happen that the mechanism design version L$_{MD}$ of a decentralized optimization problem L cannot be solved at all (even if we disregard computational considerations) – see the auction example in the previous subsection.

Definition 5.4 *Let X be some complexity class. We say that X is closed under truthfulness if for every decentralized optimization problem (language) L∈X, the mechanism design version of L is also in X.*

The example in the previous subsection shows that even if we disregard computational considerations, and allow our algorithms to run in time that is exponential in the size of the input, the truthfulness requirement alone may be sufficient to make an optimization problem hard to approximate *within any factor*. This is true even for simple and computationally-tractable (in constant time!) non-utilitarian problems like the auction setting described above. Hence, trivially classes like P and APX (those problems that can be approximated within some constant factor in polynomial time) are not closed under truthfulness.

In contrast, classical VCG theory implies that for any decentralized optimization problem L, in which the optimization goal is utilitarian, if L is in P, NP, or NP-hard, then the mechanism design version of L is also in P, NP, or NP-hard (respectively). What about APX?

Definition 5.5 *Let X be some complexity class. We define X_{util} to be subset of decentralized optimization problems in X in which the optimization goal is utilitarian.*

As explained above, VCG theory implies that P_{util}, NP_{util}, and NP-hard$_{util}$ are closed under truthfulness. *Our computational-complexity result for* CPPP *presented in this chapter essentially states that APX_{util} is NOT closed under truthfulness* (our communication-complexity result proves an analogous statement in the communication model).

5.5 Discussion and Open Problems

What are the limitations of our techniques? It would be very interesting to explore for which mechanism design problems one can prove characterizations similar to that shown in Lemma 5.1 (for the CPPP), thus proving unconditional lower bounds. It would also be very interesting to extend the proof technique of Theorem 5.4 to other succinctly described settings. We believe that such computational hardness results can be shown for many other mechanism design problems, such as combinatorial auctions. Finally, a big computational-complexity-related open question is proving that mechanism design is computationally intractable under weaker computational assumptions. We suspect that this is possible, but the precise way of doing this eludes us at the moment.

Chapter 6

Incentive-Compatibility Can Be Costly

6.1 Introduction

In the previous chapter we have shown that achieving truthfulness can be hard, in the sense that for some problems the truthfulness requirement leads to inherently bad solutions for optimization problems (in terms of approximation). In this chapter we take a completely different approach to proving that truthfulness is hard to obtain. Consider the following problem: We are given an algorithm, and wish to *make it incentive-compatible.* This requires us to come up with a *payment scheme* that induces truthful behaviour by the players. As explained in the previous chapter (Section 4) such a payment scheme may not even exist. However, even if it does exist, *can it be that the task of computing payments is simply too costly in terms of computation*? That is, can it be that while our original algorithm was reasonable to execute, the overhead due to the demand for incentive compatibility is unreasonable? In this chapter we establish that the overhead due to the demand for incentive compatibility can be significant. Specifically, making an algorithm incentive-compatible may lead to an increase in the computational burden that is linear in the number of players. This factor can be substantial in large-scale economic settings (like the interdomain routing setting discussed in previous chapters, electronic-commerce systems, and more).

6.1.1 The Communication Cost of Selfishness

Consider the goal of implementing algorithms in environments with self-interested players. We seek algorithms that admit the following two preliminary properties: First, *tractability*, which in this chapter we interpret in the information-theoretic sense, i.e., a low amount of information needs to be communicated in order to realize the outcome of the algorithm. Second, *incentive compatibility*, i.e., the *existence* of some payment scheme that supports the implementation of the algorithm in equilibrium. In this chapter, we show that tractability and the existence of incentive-compatible payments are insufficient to establish that implementing the algorithm in equilibrium will indeed be simple. This is due to the fact a non-trivial amount of *additional* communication between the different parties may be required in order to compute the equilibrium-supporting prices.

The question of how much overhead one incurs from the computation of incentive-compatible payments was recently introduced by Fadel and Segal [28], who termed this overhead the *com-*

munication cost of selfishness. In their paper, they studied the communication overhead both for Bayesian equilibria and for ex-post equilibria. In this work we focus only on ex-post equilibria - i.e., situations in which players would not want to change their behaviour in retrospect, even if they were told (after the fact) everything about the other players. Our main result shows that the amount of communication that is required to compute equilibrium-supporting prices may increase the communication-complexity of the algorithm by a factor that is linear in the number of players.

Theorem: [Informal] *There are social-choice functions such that their outcome can be computed by communicating x bits, but determining both the outcome* and *equilibrium-supporting prices may require about $n \cdot x$ bits of communication, where n is the number of players.*

We prove that this result holds even for very simple single-parameter domains, where in each possible outcome every player either "wins"or "loses". The theorem is proven under the common normalization assumption, which in the single-parameter domain we consider simply means that "losers" pay zero. While this assumption does not seem very restrictive at first glance, we currently do not know how to relax it. Whether a similar result can be proven without the normalization assumption is left as an open question. Our lower bound is the first evidence that the communication overhead due to the demand for incentive compatibility may be significant. This linear factor in the number of players may become substantial in large-scale economic settings (large networks, electronic-commerce systems).

6.1.2 Challenges

Informally, in order to prove our main result we needed to construct a social-choice function f for which the following requirements hold:

1. f can be implemented in ex-post equilibrium (in single-parameter domains this means that f should be monotone, see Section 6.2).
2. f can be computed with low communication complexity.
3. Computing the equilibrium-supporting prices requires high communication complexity.

The difficulty in finding such a social-welfare function is demonstrated by contrasting such desirable functions with two classic economic problems, *public goods* and *single-item auctions.* For these problems, we show that the requirements are *not* met. This enables us to prove upper bounds for these two problems, by showing that that the additional information required to compute the equilibrium-supporting prices is *low* (up to a small constant multiplicative factor). This claim is proven in an inherently different way for each one of these problems.

Public goods: Consider a social planner who wants to know whether a bridge should be built or not. A set of players have privately known utilities from using the bridge $v_1, ..., v_n$ and the bridge should be built only if $\sum_{i=1}^{n} v_i \geq C$ where C is its construction cost. We prove that computing the outcome plus the payments merely requires about three times the communication requirements of computing the outcome alone. Hence, the overhead in this case is small.

Single-item auction: In a single-item auction, players have private values $v_1, ..., v_n$ for the item on sale, and our goal is to sell the item to the player with the highest value. We prove that determining the right allocation and the appropriate payments requires at most three times the communication needed to determine the allocation alone. For the special case of $n = 2$, we prove an even better upper bound. The 2-player problem turns out to be equivalent to the following interesting communication complexity problem: There are 2 players, each holding a number represented by k bits. What is the communication complexity of computing the minimum of these two numbers (such that both players will know the result)? While a proving an upper bound of $k + O(\sqrt{k})$ is fairly easy, we improve this bound to $k + O(\log k)$.

As these two problems illustrate, coming up with a social-welfare function for which all of our requirements hold is a non-trivial task. We stress that achieving a better lower bound than the n lower bound shown in this chapter may be hard. This is due to the fact that it is likely to involve the construction of multi-parameter non-welfare-maximizing (non-utilitarian) social-choice functions, a class of functions that is little understood.

6.1.3 Related Work

Fadel and Segal [28] were the first to study the communication overhead of incentive compatibility. They proved exponential upper bounds, both for Bayesian-Nash equilibria and an ex-post equilibria. They presented a surprising matching exponential lower bound for the Bayesian case, but their only lower bound for the ex-post equilibrium case was 1 extra bit. The main open question posed in their paper remains unsolved: can the exponential upper bound for the communication overhead in ex-post implementation be matched by a lower bound?[1] [28] also proves a linear (in the number of players) upper bound on the communication overhead of incentive compatibility in single-parameter domains. Both [28] and our work belong to a more general line of research studying communication and information aspects of various economic environments, for example in auctions [79, 11] and in more general economic domains [10] (see [93] for a recent survey in the context of combinatorial auctions). The basic model for communication complexity was presented by Yao [102]. A survey on communication complexity can be found in [60].

6.1.4 Organization of the Chapter

The rest of the chapter is organized as follows: We present our model and notations in Section 6.2. We prove a constant upper bound on the informational overhead of incentive-compatibility for the classic model of single-item auctions in Section 6.3. In Section 6.4 we prove our results for public goods problems. We present a construction that proves a linear lower bound in Section 6.5.

6.2 Background and The Model

6.2.1 Mechanism Design

The mechanism design setting considered in this chapter is as follows: There n players, and a set of outcomes (alternatives) O. Each player i has a *valuation function* (sometimes simply referred to as "valuation") $v_i : O \to \mathbb{R}_{\geq 0}$, that belongs to a set of valuation functions V_i (a valuation functions *space*). A *social-choice function* (SCF) is a function that assigns every n-tuple of players' valuation

[1] This question appears to be hard to solve. The overhead in known to be at most linear (in the number of players) for welfare-maximization objectives and in single-parameter domains [28]. In other (multi-dimensional, arbitrary objectives) domains, the space of implementable social-choice function is not well understood. Therefore, constructing such a negative example is hard.

functions $v = (v_1, ..., v_n) \in V_1 \times \ldots \times V_n$ (valuations *profile*) an outcome $o \in O$. Each v_i is private and only known to i.

A *payment function* is a function $p : V_1 \times \ldots \times V_n \rightarrow \mathbb{R}^n$.

Definition 6.1 *A social-choice function f is said to be* implementable *(in the ex-post Nash sense) if there is a payment function p such that the following holds:*

$$\forall v = (v_1, ..., v_n) \in V_1 \times \ldots \times V_n,\ \forall i \in [n],\ \forall v'_i \in V_i,$$

$$v_i(f(v)) - p(v) \geq v_i(f(v'_i, v_{-i})) - p(v'_i, v_{-i})$$

(where (v'_i, v_{-i}) is the type profile in which i has type v'_i and every player $j \neq i$ has type v_j)

Informally, f is implementable if it is possible to come up with a payment scheme that incentivizes players to report truthful information. For example, the social-choice function in single-item auctions (where the item should be sold to the player with the highest value) is $f(v_1, ..., v_n) \in \arg max_{i \in [n]} v_i$ (breaking ties lexicographically), where v_i is the value of agent i for the item. The payment scheme in a *second-price* (Vickrey) auction is known to implement this social-choice function in equilibrium.

In this chapter we provide several examples of single-parameter domains, where the type on a player can be represented by a single scalar. We consider specific single-parameter environments where every outcome defines whether each player "wins" or "loses" ($f(v) \subseteq [n]$ is the set of winners). The player gains a value of $v_i \geq 0$ if he wins, and he gains 0 when losing. We focus on normalized mechanisms in which losers pays 0. The following is a well known characterization of implementable single-parameter social-choice functions.

Definition 6.2 *A single parameter SCF f is* monotone *if for every player $i \in [n]$, all $v_{-i} \in V_{-i}$ and all $v'_i > v_i$ s.t. $v'_i, v_i \in V_i$ it holds that if $i \in f(v_i, v_{-i})$ then $i \in f(v'_i, v_{-i})$.*

The following observation is well known.

Observation 6.1 *A single parameter SCF f is implementable if and only if it is monotonic. In the case of normalized mechanisms a winner has to pay the minimal bid she has to declare in order to win.*

6.2.2 Communication Overhead of Incentive Compatibility

We consider the communication problem in which each v_i is private and only known to i, and the players need to exchange information in order to compute the outcome of f. We work in the broadcast ("number on the forehead") model in which each sent bit is received by all players (and not addressed only to one player). Let $CC(f)$ denote the communication complexity of computing the outcome of f (see [60] for an introduction to communication complexity). Informally, the communication complexity of a function is the minimal number of bits that is required to compute the function (for any input).

How much additional communication burden is imposed by the necessity to compute payments that guarantee truthfulness? Let $CC(f, p)$ denote the communication complexity of computing the outcome of f and payments that guarantee incentive-compatibility (that is, computing the outcome of both f and *some* payment function that leads to the implementability of f.).

In order to formally define the informational overhead of incentive-compatibility we need to be concrete about the information each player holds: Let f_k be a social-choice function with n players such that each player's valuation is represented using k bits of information, for some fixed $k \in \mathbb{N}$. Formally, $f_k : \{0,1\}^{k \times n} \to O$, i.e., for every $(v_1, v_2, ..., v_n)$ with each $v_i \in \{0,1\}^k$, f_k picks an outcome $f_k(v_1, v_2, ..., v_n)$.

Definition 6.3 *The* informational overhead of incentive-compatibility (IOIC) *of f_k is defined to be $IOIC(f_k) = \frac{CC(f_k,p)}{CC(f_k)}$.*

Our main result shows that for some social-choice functions f_k a significant informational overhead may be incurred, and we prove that this holds even in single parameter domains. Formally, in such domains the valuation of a player is given by a number in $[0,1]$ represented by k bits (k is the precision in the representation of v_i). That is, for every player i there is some $t_i \in \{0, ..., 2^k - 1\}$, such that $v_i = t_i \cdot 2^{-k}$.

6.2.3 Communication Complexity: Background and Basic Observations

This section presents some basic background of some of the tools we use from the theory of communication complexity. For a comprehensive survey on the subject we refer the reader to [60].

Observation 6.2 *If the range of function f is of size m ($|Range(f)| = m$), then any communication protocol for f requires at least $\log(m)$ bits.*

Observation 6.3 *For any implementable function f_k with n players each holding k bits it holds that $CC(f_k) \leq k(n-1) + \lceil \log(|Range(f_k)|) \rceil$ and $CC(f_k, p) \leq kn$ ($Range(f_k)$ is the set of different outcomes in the range of f_k).*

Proof: This is shown by considering two trivial protocols: To compute f_k, each of the players but the last one transmits all his information, and the last player computes the outcome and transmits the outcome. As there are $|Range(f_k)|$ possible relevant outcomes, $\lceil \log(|Range(f_k)|) \rceil$ bits are clearly sufficient to encode all these outcomes. To compute $CC(f_k, p)$ simply let all players transmit all of their private information. ∎

Our proofs use a common communication-complexity technique called *fooling sets*. Intuitively, a fooling set is a large set of possible inputs such that any communication protocol must be able to distinguish between every two of them. Fooling sets arguments are based on following well known property of communication protocols. If a protocol executes exactly the same on two inputs, it must do so on any possible "combinations" of these inputs, thus must output the same outcome. For completeness, we give provide a brief definition of the fooling-set technique below. More details can be found in the textbook [60].

Fooling sets. Suppose that some communication protocol that computes a social function f for a two-player setting cannot distinguish between (v_1, v_2) and (v_1', v_2'), then, that protocol cannot also tell the difference between these inputs and (v_1', v_2), (v_1, v_2'). This suggests a way for proving lower bounds on the communication complexity of social-choice functions: Find a subset of the inputs and prove that every member in this subset is assigned the same outcome, but a combination of *every* two members is assigned a different outcome. This will imply that *any* communication protocol *must* distinguish between *every* two members of the subset of inputs. This, in turn, would mean that the logarithmic value of the cardinality of this subset is a lower bound on the number of bits that need to be transmitted in order to compute f. Formally, let f be a social-choice function. Let v, v' be two valuation functions. Let

$$V_{v,v'} = \{v'' = (v''_1, ..., v''_n) | \forall i \in [n] \; v''_i = v_i \; or \; v''_i = v'_i\}$$

A well known fact in communication complexity (see [60]) is the following:

Theorem 6.1 [Fooling Set Argument] *Let $f : V = V_1 \times \ldots \times V_n \rightarrow O$ be a social-choice function. For every $V' \subseteq V$ such that:*

- *there is an outcome $o^* \in O$ such that for every $v' \in V'$ $f(v') = o^*$.*
- *for every $v, v' \in V'$ $\exists v'' \in V_{v,v'}$ such that $f(v'') \neq o^*$*

it holds that $CC(f) \geq log(|V'|)$.

6.3 Overhead in Single-Item Auctions

A well known economic setting is the single-item auction, where a seller aims to sell an item to the bidder who values it the most.

Definition 6.4 (SINGLE-ITEM-AUCTION)
Input: valuations $v_1, ..., v_n \in \mathbb{N}$
Output: a bidder with the highest value, i.e., $\arg max_{i \in [n]} v_i$ (breaking ties lexicographically).

If bidder i wins his value for the outcome is v_i, otherwise his value for the outcome is 0. From SINGLE-ITEM-AUCTION we can derive the function SINGLE-ITEM-AUCTION$_k$ for the case that $v_i = t_i \cdot 2^{-k}$ for some integer $t_i \in \{0, ..., 2^k - 1\}$ and is represented by a k-bit string. It is well known that if the winner pays the second highest price then the auction is truthful.

Next we show that for SINGLE-ITEM-AUCTION the informational overhead of incentive-compatibility is at most a small constant (3).

Proposition 6.1 *The informational overhead of incentive-compatibility for the social-choice function* SINGLE-ITEM-AUCTION *is at most 3.*

Proof: To prove the claim we show that for every k, it holds for the social-choice function f_k =SINGLE-ITEM-AUCTION$_k$ that $CC(f_k, p) \leq 2 \cdot CC(f_k) + k \leq 3 \cdot CC(f_k)$. The last weak inequality is a result of the following claim:

Claim 6.1 *For every $n \geq 2$, CC(*SINGLE-ITEM-AUCTION$_k$*)$\geq k$*

Proof: We shall prove that $CC(f_k)$ is large by constructing a "fooling set" of size 2^k. By Theorem 6.1 this shows that $CC(f_k) \geq k$. Consider all pairs (v_1, v_2) such that $v_1 = v_2$. No two such pairs can be mapped by a protocol that computes f to the same monochromatic rectangle (exactly the same execution of the protocol which leads to the same outcome). Let (v_1, v_2) and

(v'_1, v'_2) be two such type-profiles that are mapped to the same rectangle. Observe, that for all these type-profiles bidder 1 wins and bidder 2 loses. W.l.o.g., let $v_1 < v'_1$. Then, (v_1, v'_2) should also be mapped to the same rectangle. However, this leads to a contradiction because in this case player 2 should win. Hence, there are at least 2^k rectangles, and so any protocol that computes f must transmit at least k bits. ∎

To show that $CC(f_k, p) \leq 2CC(f_k) + k$ we present a simple protocol for $CC(f_k, p)$: run the protocol for f_k to find the player with highest value. Remove the highest bidder and run the protocol for f_k again. Now the players know who is the player with the second highest value. Finally, this player transmits his value, which requires k more bits. ∎

6.3.1 An improved analysis for $n = 2$

Auctioning a single item is probably the most fundamental problem in mechanism design. In this section we present a better analysis of the communication burden in second-price auctions with two players. We present an iterative mechanism where players broadcast their information in turns, and an alphabet-changing trick that allows us to save in information (a similar protocol without changing alphabets proves a $1 + O(\sqrt{k})$ bound).

Theorem 6.2 *For the social-choice function* SINGLE-ITEM-AUCTION$_k$ *and* $n = 2$*, the information overhead of incentive compatibility is at most* $1 + O(\frac{\log(k)}{k})$.

Theorem 6.3 *For the social-choice function* SINGLE-ITEM-AUCTION$_k$ *and* $n = 2$*, the information overhead of incentive compatibility is at most* $1 + O(\frac{\log(k)}{k})$.

Proof:

By Observation 6.1 proved in the proof for Proposition 6.1, $CC(f_k) \geq k$. The theorem is a direct result of the following communication-complexity lemma. In the terms of our model, it shows that for $n = 2$, $CC(\text{SINGLE-ITEM-AUCTION}_k, p) \leq k + O(\log(k))$ (recall that in single-item auctions, an equilibrium-supporting price is well known to be the second-highest price).

Lemma 6.1 *Consider two players* 1 *and* 2*, each holds a number represented by* k *bits* v_1, v_2 *(resp.). Both players can realize* $\min\{v_1, v_2\}$ *with communication of at most* $k + O(\log(k))$ *bits.*

Proof: We present a protocol that computes f_k and incentive-compatible payments, and requires the transmission of at most $k + O(\log(k))$ bits. It is well known that if the winning bidder is charged the value of the losing bidder then incentive-compatibility is guaranteed (this is the celebrated "second-price auction"). So, computing f_k and incentive-compatible payments can be done by finding the identity and value of the bidder with the lowest value.

We consider the following communication protocol: Let $m = \lceil \log(k) \rceil$. The two parties (bidders) encode their inputs (values) using an alphabet of size $2^m - 1$. The two parties now take turns sending the symbols ("blocks") of their input values (from most significant to least significant). The alphabet symbols are encoded using m bits and are represented by the strings 0^m through $1^{m-1}0$. The string 1^m is assigned a reserved meaning: "your previous string was not equal to mine". If one party sends the 1^m string, then it also sends an additional bit indicating who has the larger value, and that party then sends all of its remaining symbols (encoded normally).

What is the maximal cost of this protocol in bits? We are wasting at most two blocks and one more bit when one party discovers the differing blocks (the most wasteful possibility is if the non-sending party realizes that its input is smaller, in which case he needs to send the string with the reserved meaning, an additional bit indicating that he has the smaller value, and repeat the last symbol). The cost is therefore at most $m \times R + 2m + 1$, where R is the length of the k-bit input encoded in the alphabet of size $2^m - 1$. That is, the number of symbols (blocks) sent times the number of bits needed to represent each symbol, plus the number of bits in two additional blocks, and another bit. The size of R must be such that $(2^m - 1)^R \geq 2^k$, that is, we can set R to be $\lceil k/\log(2^m - 1)\rceil$. Hence, the total cost in bits (setting $m = \lceil \log(k) \rceil$) is at most $\lceil \log(k) \rceil \times \lceil k/\log(k-1) \rceil + 2\lceil \log(k) \rceil + O(1)$, which is $k + O(\log(k))$. ∎

The theorem follows. ∎

We note that the proof of this result implies that the informational overhead of of incentive-compatibility for Single-Item-Auction with $n = 2$ is extremely small and tends to 1 very fast with k.

6.4 Overhead in the Public-Good Model

We now consider another classic economic setting - the construction of a public project ("public good"). Each player in a set of players has a "benefit" of v_i from using the public good, and the social planner aims to build it only if the sum of benefits exceeds the construction cost C. The function is defined given the parameter $C \geq 0$.

Definition 6.5 (C-Public-Good)
Input: valuations $v_1, ..., v_n \in \mathbb{N}$.
Output: "Build" if $\sum_{i=1}^{n} v_i \geq C$, *"Do not build" Otherwise.*

It is easy to observe that the payments that implement this SCF in a normalized mechanism are $p_i = C - \sum_{j \neq i} v_j$ in the case of "Build" (and all players win). Again, we consider the derived SCF C-Public-Good$_k$ for the case that $v_i = t_i \cdot 2^{-k}$ for some integer $t_i \in \{0, ..., 2^k - 1\}$ and is represented by a k-bit string.

In the next section we consider the problem for the case of 2 agents. We show that the informational overhead of incentive-compatibility for that case is almost 2. Yet, in the following section we show that when moving to an n-player setting the overhead remains constant and does not grow with n.

6.4.1 A Lower Bound for 2-Player Settings

We start by considering the case of only 2 players and show that there is lower bound that is almost 2 on the informational overhead of incentive-compatibility.

Proposition 6.2 *Assume $n = 2$. For any $\epsilon > 0$, for any large enough k there exists a cost C such that the informational overhead of incentive-compatibility for the social-choice function C-Public-Good$_k$ is at least $2 - \epsilon$.*

Proof: To prove the claim we consider the SCF f_k =C-Public-Good$_k$ for the case that the cost C of the public good is $1 - 2^{-k}$.

By Observation 6.3 it holds that $CC(f_k) \leq k + 1$ (as there are only 2 possible outcomes $|Range(f_k)| = 2$). In order to prove the proposition we show below that $CC(f_k, p) \geq 2k - 1$. This implies that the informational overhead of incentive-compatibility of C-PUBLIC-GOOD$_k$ is at least $\frac{2k-1}{k+1} = 2 - \frac{3}{k+1}$. Clearly for any $\epsilon > 0$ there is a k such that this is larger than $2 - \epsilon$. To conclude the proof of the theorem we next show that $CC(f_k, p) \geq 2k - 1$.

Claim 6.2 *When $n = 2$ and $C = 1 - 2^{-k}$, $CC(C\text{-PUBLIC-GOOD}_k, p) \geq 2k - 1$.*

Proof:

Figure 6.1 describes the function f_k. We shall call any pair of possible values (v_1, v_2) a *type-profile*. Observe, that there are 2^{2k} possible type-profiles, out of which $2^{2k-1} + 2^{k-1} > 2^{2k-1}$ are type profiles in which $v_1 + v_2 \geq C$. We shall prove that for every two type-profiles $v = (v_1, v_2)$ and $v' = (v'_1, v'_2)$, such that $v_1 + v_2 > C$, $v'_1 + v'_2 > c$, and $v \neq v'$, it must hold that any protocol that computes incentive-compatible payments outputs $p(v_1, v_2) = (p_1(v_1, v_2), p_2(v_1, v_2)) \neq p(v'_1, v'_2) = (p'_1(v_1, v_2), p'_2(v_1, v_2))$. This would imply that f_k has at least 2^{2k-1} different outcomes in its range, which means that at least $log(2^{2k-1}) = 2k - 1$ bits must be transmitted (by Observation 6.2).

Let $v = (v_1, v_2)$ and $v' = (v'_1, v'_2)$ be two type-profiles such that $v_1 + v_2 > C$, $v'_1 + v'_2 > C$, and $v \neq v'$ (different type-profiles in which both players win). W.l.o.g. assume that $v_1 \neq v_2$. Then, we shall show that $p_2(v_1, v_2) \neq p_2(v'_1, v'_2)$. As shown in Figure 6.1, the payment of a winning player must be the minimal value it needed in order to win. That is, the payment of player 1 is the horizontal projection on the diagonal line, and the payment of player 2 is the vertical projection on the diagonal line. Hence, if the value of player 1 is v_1 then the payment of player 2, $p_2(v_1, v_2)$, is exactly $C - v_1$ (that is, the minimal value of player 2 for which they both win). Similarly, $p_2(v'_1, v'_2) = C - v'_1$. Since $v_1 \neq v'_1$ we conclude that $p_2(v_1, v_2) \neq p_2(v'_1, v'_2)$. ∎

The theorem follows. ∎

6.4.2 A Constant Upper Bound for n-Player Settings

As we have seen, for $n = 2$ the informational overhead of the public good function is essentially 2.[2] One might hope that a generalization of this function to an n-player setting leads to a lower bound of n. Yet we show that this is not the case.

Theorem 6.4 *Fix $\epsilon > 0$. For any C, the informational overhead of incentive-compatibility of the social-choice function C-PUBLIC-GOOD$_k$ with $n \geq 3$ agents and k that is large enough is at most $2 \cdot \frac{n}{n-1} + \epsilon \leq 3 + \epsilon$.*

Proof: The players values $v_1, ..., v_n$ are all in $[0, 1]$ and we are interested in figuring out whether $\Sigma_i v_i \geq C$. Obviously in $C = 0$ or $C \geq n$ then the answer is trivial. So, we can consider the case $C \in (0, n)$. For any integer k' that is large enough, we can pick k such that $C \cdot 2^k \in [2^{k'-1}, 2^{k'}]$. We now consider the function f_k and normalize the input by 2^k, that is, we think about the input as if each agent i has integer value $v_i \in \{0, ..., 2^k - 1\}$ and the cost C is $\lceil C \cdot 2^k \rceil \in [2^{k'-1}, 2^{k'}]$. (note that as the values are now integers we can round up $C \cdot 2^k$ while being left with the same decision problem). The communication complexity of the original problem is clearly equivalent to the communication complexity of the new problem after normalization.

It is easy to see that $CC(f_k, p)$ is at most $nk' + n$, because of the following protocol: Ask each player if his value is at least C (this requires n bits). Ask all the players whose values are lower

[2] It is relatively easy to derive a matching upper bound of about 2 as it is clear that $2k$ bits are always sufficient to compute the payments and it is relatively easy to show that k bits are necessary to compute the outcome.

than C to transmit their v_i's (this requires k' bits per player, i.e., at most $n \times k'$ bits). An easy observation is that this simple protocol provides us with sufficient information to calculate the prices for all players. We shall now prove a lower bound on $CC(f_k)$ that will imply the theorem.

We prove the following lemmas:

Lemma 6.2 *If $C \leq \frac{n}{2}$, then for large enough k' the communication complexity of determining whether $\Sigma_i v_i \geq C$ is at least $(\frac{n}{2}-1)\cdot k' - f(n)$ for some function $f(\cdot)$.*

Proof: We shall construct a large fooling set and invoke Theorem 6.1. Consider all the possible type-profiles $(v_1,\ldots,v_n)$ such that $\Sigma_i v_i = C$. Obviously for all these type-profiles $\Sigma_i v_i \geq C$. However, any protocol that tries to determine whether $\Sigma_i v_i \geq C$ cannot place any two such type-profiles in the same monochromatic rectangle (see [60]) for the following reason: Let $v = (v_1,\ldots,v_n)$ and $v' = (v'_1,\ldots,v'_n)$ be two different such type-profiles. Let j be a coordinate such that $v_j \neq v'_j$. W.l.o.g, assume that $v_j < v'_j$. Then, if v and v' are in the same rectangle then so is the type profile $v'' = (v''_1,\ldots,v''_n)$ in which $v''_i = v'_i$ for every $i \neq j$ and $v''_j = v_j$. However, this leads to a contradiction because $\Sigma_i v''_i < C$ (since the outcome of the protocol cannot be the same for v and v'').

Hence, by finding a lower bound L on the number of possible type-profiles $(v_1,\ldots,v_n)$ such that $\Sigma_i v_i = C$, we also find a lower bound on the number of monochromatic rectangles of any protocol that computes C-PUBLIC-GOOD$_k$. This implies that $\log L$ is a lower bound on the number of bits transmitted by any such protocol. We reach L as follows: First, consider the case that $C \leq 2^k - 1$. We consider the following family of type-profiles: $v_1 = C - 2^{k'-1}, v_2 = ... = v_{\frac{n}{2}-1} = 0$, and $\Sigma_{i=\frac{n}{2}}^{n} v_i = 2^{k'-1}$. Observe, that any type-profile in this family is such that $\Sigma_{i=1}^{n} v_i = C$. How many such type-profiles are there? There are $\binom{2^{k'-1}+\frac{n}{2}}{\frac{n}{2}}$ ways to distribute $2^{k'-1}$ between $\frac{n}{2}+1$ players. This is bounded from below by $\frac{2^{(k'-1)\frac{n}{2}}}{(\frac{n}{2})^{\frac{n}{2}}}$. So, any protocol that determines whether $\Sigma_i v_i \geq C$ needs to transmit at least $\log(\frac{2^{(k'-1)\frac{n}{2}}}{(\frac{n}{2})^{\frac{n}{2}}}) = \frac{n}{2}k' - \frac{n}{2}(\log(n))$ bits.

What if $C \geq 2^k - 1$? Recall that $C \leq \frac{n}{2}\cdot 2^k$ (after the normalization) so in this case observe that for large enough k we can distribute $C - (2^k - 1)$ to players $1,...,\frac{n}{2}$ (in some arbitrary way). This is so as these $\frac{n}{2}$ agent can split up to $(2^k-1)\cdot\frac{n}{2}$, and for large enough k it holds that $(2^k-1)\cdot\frac{n}{2} \geq 2^k\cdot\frac{n}{2} - (2^k-1) \geq C - (2^k-1)$. So now we are left with a cost of $C' = 2^k - 1$ to distribute between agents $\frac{n}{2}+1,...,n$. We can now achieve L by looking at the different ways to distribute C' between players $\frac{n}{2}+1,...,n$, i.e., such that $\Sigma_{i=\frac{n}{2}+1}^{n} v_i = C'$. How many such type-profiles are there? There are $\binom{C'+\frac{n}{2}-1}{\frac{n}{2}-1}$ ways to distribute C' between $\frac{n}{2}$ players. Now

$$\binom{C'+\frac{n}{2}-1}{\frac{n}{2}-1} = \binom{(2^k-1)+(\frac{n}{2}-1)}{\frac{n}{2}-1} \geq \frac{(2^k-1)^{(\frac{n}{2}-1)}}{(\frac{n}{2}-1)^{(\frac{n}{2}-1)}}$$

So, any protocol that determines whether $\Sigma_i v_i \geq C$ needs to transmit at least a logarithmic factor of this number, which is $(\frac{n}{2}-1)\, log(2^k-1) - (\frac{n}{2}-1)\log(\frac{n}{2}-1)$. Note that $\frac{n}{2}\cdot 2^k \geq C \geq 2^{k'-1}$ thus $2^k \geq \frac{2^{k'}}{n}$. We conclude that this number is at least

$$(\frac{n}{2}-1)\log(\frac{2^{k'}}{n}-1) - (\frac{n}{2}-1)(\log(n)-2)$$

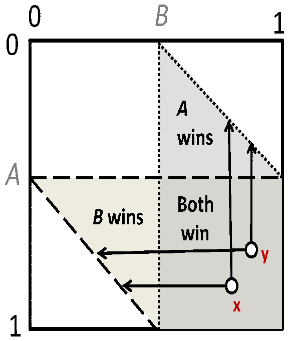


Figure 6.1: The description of the 2-Not-Too-Far social-choice function. For every profile of values for the players, the figure shows whether A wins, B wins or both. In all other profiles both lose. The hardness of this example is due to the fact that every two profiles of values for which the two players win (e.g., x and y in the picture) is associated with different prices. Note that the prices are determined by a projection on the diagonals defined by the social-choice function.

$$\geq (\frac{n}{2} - 1)k' - (\frac{n}{2} - 1)(2\log(n) - 1)$$

■

Lemma 6.3 *If $C \leq \frac{n}{2}$, then for large enough k' the communication complexity of determining whether $\Sigma_i v_i \leq C$ is at least $(\frac{n}{2} - 1) \cdot k' - f(n)$ for some function $f(\cdot)$.*

Proof: The proof of this lemma is almost identical to that of the previous one (it involves the construction of the very same fooling set). ■

The theorem will now follow because if $C \geq \frac{n}{2}$ in the original formulation then Lemma 6.2 implies that $CC(f,p)$ is at least $(\frac{n}{2} - 1) \cdot k' - f(n)$ for some function $f(\cdot)$. If $C > \frac{n}{2}$ then the problem is equivalent to figuring out whether $n - \Sigma_i v_i \geq n - C$. That is, it is equivalent to the problem in which every player has the value $1 - v_i$ and the players are trying to figure out whether the sum of their values is at most $n - C$. This problem is just as hard as the original problem, as shown by Lemma 6.3. By plugging in the values of $CC(f_k, p)$ and $CC(f,p)$ we get an IOIC of at most $\frac{nk'+n}{(\frac{n}{2}-1)\cdot k' - f(n)}$, which converges to $2\frac{n}{n-1}$ as k' goes to infinity. ■

6.5 Main Result: A Linear Lower Bound

In order to prove a lower bound of n we must identify a social-choice function f such that $CC(f)$ is substantially (essentially factor n) smaller that $CC(f,p)$.

6.5.1 Another Lower Bound for 2 Players

The reader may expect a straightforward generalization of the 2-player public-good problem to enable us to get an $\Omega(n)$ lower bound for general single-parameter domains. However, as shown in Section 6.4.2 this is not the case. In fact, the n-player public-good problem is such that the communication cost of selfishness is never greater than $3+\epsilon$.

We will now present a construction that does extend to n players, and thus obtains a linear lower bound. We will start by presenting the construction and the proof for 2 players, and then we will describe the genera construction and proof.

Consider the 2-player social-choice function depicted in Figure 6.1. As before, we have 2 players $1,2$, each holding a value $v_i = t_i \cdot 2^{-k}$ for an integer $t \in \{0,...,2^k-1\}$, represented by a k-bit string. Player i's utility from winning is v_i, and his utility from losing is 0. Player 1 wins if and only if $v_2 \geq 1/2$ and $v_1 \geq v_2 - 1/2$. Similarly, player 2 wins if and only if $v_1 \geq 1/2$ and $v_2 \geq v_1 - 1/2$. We shall refer to f_k as the "NOT-TOO-FAR$_k$" social-choice function. It is easy to check that NOT-TOO-FAR$_k$ is monotone and thus its informational overhead of incentive-compatibility is finite.

Proposition 6.3 *For any $\epsilon > 0$, for any k large enough the informational overhead of incentive-compatibility for the social-choice function* NOT-TOO-FAR$_k$ *is at least* $2-\epsilon$.

Proof: Let $f_k =$NOT-TOO-FAR$_k$. By Observation 6.3 it holds that $CC(f_k) \leq k+2$ as there are 4 outcomes in the range (any subset of the player can win).

We show below that $CC(f_k,p) \geq 2k-2$. From this we derive that $\frac{CC(f_k,p)}{CC(f_k)} \geq \frac{2k-2}{k+2} = 2 - \frac{6}{k+2}$. Clearly as this a monotonic function of k that converge to 2, for any $\epsilon > 0$ there is a k such that this is larger than $2-\epsilon$. We next derive the promised lower bound on $CC(f_k,p)$.

Claim 6.3 *For $n = 2$, $CC($*NOT-TOO-FAR$_k$*$,p) \geq 2k-2$.*

Proof: Consider the type-profiles (v_1,v_2) such that both v_1 and v_2 are at least $1/2$. Observe, that there are exactly 2^{2k-2} such type-profiles, and that both players win for each such type profile.

We shall prove that for every two such type-profiles $v = (v_1,v_2)$ and $v' = (v'_1,v'_2)$ it must hold that any communication protocol that computes incentive-compatible payments outputs $p(v_1,v_2) \neq p(v'_1,v'_2)$. Therefore, any such protocol has at least 2^{2k-2} outcomes in its range. By Observation 6.2 this implies that the minimal number of bits that must be transmitted by any such protocol is at least $\log(2^{2k-2}) = 2k-2$.

W.l.o.g., assume that $v_1 \neq v'_1$. From Figure 6.1 one can deduce that in the event that both players win the payment of each is the other's player's value minus $1/2$. We shall show that $p_2(v_1,v_2) \neq p_2(v'_1,v'_2)$. Clearly $p_2(v_1,v_2) = v_1 - 1/2$ (the minimal value for which 2 would win). Similarly, $p_2(v'_1,v'_2) = v'_1 - 1/2$. Since $v_1 \neq v'_1$ we conclude that $p_2(v_1,v_2) \neq p_2(v'_1,v'_2)$. ∎

The theorem follows. ∎

6.5.2 A Linear Lower Bound for n Players

We are now ready to prove the main theorem in this chapter. The social-choice function for which we shall prove a lower bound of about n is an extension of 2-NOT-TOO-FAR$_k$ to n-player settings.

Definition 6.6 (NOT-TOO-FAR$_k$)
Input: valuations $v_0,...,v_{n-1} \in \mathbb{N}$ represented by strings of k bits.
Output: A set of winning players from $[n]$. Player i wins if one of the following happen:

1. *$v_j \geq 1/2$ for every $j \in [n]$.*

2. *$v_j \geq 1/2$ for every $j \in [n], j \neq i$,* and *$v_i \geq v_{i+1\ (mod\ n)} - 1/2$.*

We first observe that the function Not-Too-Far$_k$ is implementable, and therefore the informational overhead of incentive compatibility is well defined. Indeed, it is easy to see that if player i wins in Not-Too-Far$_k$ and increases its bid, i will still win.

Observation 6.4 *The social-choice function* Not-Too-Far$_k$ *is monotone.*

We are now ready to present the main result, a linear lower bound on the informational overhead of incentive-compatibility for a single-parameter SCF. This is done by showing that computing both the function and the payments (CC(f,p)) requires lots of communication since there are many different price-vectors the mechanism should distinguish between; also, computing the function alone (CC(f)) requires a low amount of communication since after all players declared if their value exceeds 1/2 using 1 bit each, the only relevant information is held by up to two players (v_i and v_{i+1} for some i).

Theorem 6.5 *For any $\epsilon > 0$ and for k large enough, the informational overhead of incentive-compatibility for the social-choice function* Not-Too-Far$_k$ *is at least $n - \epsilon$.*

Proof: Let f_k =Not-Too-Far$_k$. We show below that $CC(f_k, p) \geq n(k-1)$ (Claim 6.4) and that $CC(f_k) \leq n+k+1$ (Claim 6.5). From these two facts we derive that $\frac{CC(f_k,p)}{CC(f_k)} \geq \frac{n(k-1)}{k+n+1} = n - \frac{n(n+2)}{n+k+1}$. Clearly for any $\epsilon > 0$ there is a k such that this is larger than $n - \epsilon$. We next derive the promised bounds on $CC(f_k, p)$ and $CC(f_k)$.

Claim 6.4 *$CC($*Not-Too-Far$_k$*$, p) \geq n(k-1)$.*

Proof:

There are $2^{n(k-1)}$ type-profiles $(v_0, \ldots, v_{n-1})$ such that each v_i is at least 1/2. For all these type-profiles all players win. Let $(v_0, \ldots, v_{n-1})$ and $(v'_0, \ldots, v'_{n-1})$ be two different such type-profiles. Let $p(v_0, \ldots, v_{n-1})$ and $p(v'_0, \ldots, v'_{n-1})$ be the incentive-compatible payments outputted by a communication protocol for these two type-profiles. We shall show that these two payment vectors must be different. This is derived from the fact that $p_i(v_0, \ldots, v_{n-1}) = v_{i+1\ (mod\ n)} - 1/2$, and similarly, $p_i(v'_0, \ldots, v'_{n-1}) = v'_{i+1\ (mod\ n)} - 1/2$. Hence, if any coordinate $j \in \{0, 1, ..., n-1\}$ is such that $v_i \neq v'_i$ this implies that $p_{j-1}(v_0, v_2, ..., v_{n-1}) \neq p_{j-1}(v'_0, v'_2, ..., v'_{n-1})$. Therefore, any protocol that computes incentive-compatible payments has at least $2^{n(k-1)}$ outcomes in the range of f_k and thus requires at least $n(k-1)$ bits (by Observation 6.2). We conclude that $CC(f_k, p) \geq n(k-1)$.
∎

Claim 6.5 *$CC($*Not-Too-Far$_k$*$) \leq n + k + 1$.*

Proof:

We show that $CC(f_k) \leq k + n + 1$ by exhibiting a communication protocol that computes f_k and only requires $k+n+1$ bits: First, each player i transmits a single bit b_i that indicates whether his value is at least 1/2 (i transmits 1 if $v_i \geq 1/2$). If $b_i = 1$ for all i then all players win. If for two or more players $b_i = 0$ then all players lose. If there is a player j such that $b_j = 0$ and for all other players it holds that $b_i = 1$ then all other players (but j) lose. In this case, in order to determine whether player j wins, player $j + 1$ $(mod\ n)$ transmits all of his bits. Player j (who now knows $v_{j+1\ (mod\ n)}$) checks whether $v_j \geq v_{j+1\ (mod\ n)} - 1/2$ (in which case player j wins). He now broadcasts an additional bit informing the others of the result (1 indicating "I win" and 0 indicating "I lose"). Observe, that overall $k + n + 1$ bits were transmitted, and that the protocol does indeed compute NOT-TOO-FAR$_k$. So, $CC(f_k) \leq k + n + 1$. ∎

The theorem follows. ∎

Chapter 7

Conclusions

In this thesis we have presented results of two kinds: The first kind of results was obtained by analyzing Internet protocols (and abstractions of such protocols). Specifically, we have addressed convergence and incentive issues concerning the Border Gateway Protocol (BGP), that handles interdomain routing in today's Internet. Among several others, we have presented the following results:

The first non-trivial necessary condition for BGP convergence: *If there are two stable solutions in the network, to which BGP can potentially converge, then there is a timing of update messages that results in persistent route oscillations.*

This closes a long-standing open question in networking research. The fact that the uniqueness of a pure Nash equilibrium (which corresponds to a stable solution) is a necessary condition for BGP's convergence is, in our view, one example of a more general rule. We believe that the uniqueness of pure Nash equilibria is closely related to convergence of best-reply dynamics (that corresponds to BGP) in asynchronous environments. For more evidence for such connections see [78]. Understanding the operation of simple and natural game-dynamics (like best-reply dynamics) in asynchronous settings is an important research agenda (we take a first step in this direction in [78])

Identifying realistic settings in which BGP is incentive-compatible: *Protocols like S-BGP are incentive-compatible if No Dispute Wheel holds.*

This provides strong strategic justification for well-known modifications of BGP proposed by security research. Our result highlights an interesting connection between the two research agendas that handle lack of compliant behaviour in interdomain routing: security research and distributed algorithmic mechanism design. Essentially, we show that security enhancements of BGP that are well-known in the networking community are incentive-compatible.

Motivated by the fruits of our analysis of BGP we proposed a general framework for the analysis of Internet protocols like BGP (that are based on best-reply dynamics), which we call "best-reply mechanisms". This framework turns out to be very useful in the analysis of other Internet settings (congestion control, adword auctions), and various other settings (economic markets, cost-sharing problems, and more). The notion of best-reply mechanisms gives broader context, strengthens, and unifies several known results, and leads to many new ones. There are very few general techniques for the design of incentive-compatible mechanisms. In particular, the only known such technique for designing *deterministic* incentive-compatible algorithms is VCG. The many applications of best-reply mechanisms encourage us to think that they may prove to be a fairly general technique for performing such feats.

The second kind of results discussed in this thesis are impossibility results. These results address a major design requirement from protocols in economic settings (like the Internet), namely, incentive-compatibility.

Incentive-Compatibility can lead to bad approximations: *There are problems (e.g., combinatorial public projects) that are in APX, and for which an optimal incentive-compatible algorithm exists, but for which no constant approximation-ratio is achievable by algorithms that are both computationally efficient and incentive-compatible.*

This result, which is our main impossibility result, lies at the very heart of algorithmic mechanism design, and can be regarded as an attempt to connect traditional complexity theory with algorithmic mechanism design. We believe that there is still much work to be done on unifying traditional complexity theory and the rapidly developing theory of algorithmic mechanism design. Understanding the connections between complexity classes and incentive-compatibility is an important research agenda. Our second impossibility result is based on an utterly different way of evaluating the hardness of incentive-compatibility:

The overhead of computing incentive-compatible payments might be big: *There are algorithms whose outcome can be computed by communicating x bits, but for which determining both the outcome* and *incentive-compatible prices may require about $n \cdot x$ bits of communication, where n is the number of players.*

It is educating to contrast our possibility results regarding BGP's incentive-compatibility (or, more accurately, the incentive-compatibility of modifications of BGP) with our two general impossibility results, thus seeing how BGP (and other Internet protocols) avoids these impossibility barriers. Basically, our impossibility results state that in certain situations, incentive-compatibility can either harm the quality of the solutions of an algorithm (approximation-wise) or lead to high computational costs (from computing payments). These two impossibility results do not apply to BGP mainly because it is completely outside their conceptual scope. That is, we have shown that BGP is actually not about optimization at all (and that, in fact, any algorithm for social-welfare maximization in interdomain routing will perform very poorly in the worst case even in realistic settings). Hence, there is no clear notion of how to measure the quality of the routing outcomes reached by BGP. As for the overhead of computing incentive-compatible payments, we have shown that BGP (or, more precisely, modifications of BGP) achieves incentive-compatibility *without money*. Hence, in this sense, incentive-compatibility has *no* cost.

Bibliography

[1] H. Ackermann, P. Goldberg, V. S. Mirrokni, H. Roeglin, and B. Voecking. Uncoordinated two-sided markets. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 256–263, 2008.

[2] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th annual ACM symposium on Theory of computing*, pages 10–19, 1998.

[3] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. H. Papadimitriou. Selfish behavior and stability of the internet: a game-theoretic analysis of TCP. In *Proceedings of ACM SIGCOMM*, pages 117–130, 2002.

[4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–137, 1999.

[5] A. Archer, C. H. Papadimitriou, K. Talwar, and E. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proceedings of the 14th annual ACM-SIAM symposium on Discrete algorithms*, pages 205–214, 2003.

[6] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 482, 2001.

[7] M. Babaioff, L. Blumrosen, M. Naor, and M. Schapira. Informational overhead of incentive compatibility. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 88–97, 2008.

[8] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pages 72–87, 2003.

[9] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, and Arunava Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[10] L. Blumrosen and M. Feldman. Implementation with a bounded action space. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 62–71, 2006.

[11] L. Blumrosen and N. Nisan. Auctions with severely bounded communications. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 406–415, 2002.

[12] L. Blumrosen and N. Nisan. Combinatorial auctions. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 11. Cambridge University Press, 2007.

[13] R. I. Brafman and M. Tennenholtz. Efficient learning equilibrium. *Artificial Intelligence*, 159(1-2):27–47, 2004.

[14] R. I. Brafman and M. Tennenholtz. Optimal efficient learning equilibrium: Imperfect monitoring in symmetric games. In *Proceedings of the National Conference on Artificial Intelligence*, pages 726–731, 2005.

[15] K. Butler, T. Farley, P. McDaniel, and J. Rexford. A survey of BGP security issues and solutions. Technical report, AT&T Labs - Research, 2004.

[16] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. R. Karlin, C. Mathieu, and M. Schwarz. Greedy bidding strategies for keyword auctions. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 262–271, 2007.

[17] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

[18] V. Conitzer and T. Sandholm. Communication complexity as a lower bound for learning in games. In *Proceedings of the 21st international conference on Machine learning*, page 24, 2004.

[19] R. R. Dakdouk, S. Salihoglu, H. Wang, H. Xie, and Y. R. Yang. Interdomain routing as social choice. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops*, page 38, 2006.

[20] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proceedings of ACM SIGCOMM*, pages 1–12, 1989.

[21] S. Dobzinski and N. Nisan. Limitations of VCG-based mechanisms. In *Proceedings of the 39th annual ACM symposium on Theory of computing*, pages 338–344, 2007.

[22] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 610–618, 2005.

[23] S. Dobzinski and M. Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the 17th annual ACM-SIAM symposium on Discrete algorithm*, pages 1064–1073, 2006.

[24] S. Dobzinski and M. Sundararajan. On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 38–47, 2008.

[25] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

[26] A. Fabrikant and C. H. Papadimitriou. The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In *Proceedings of the 19th annual ACM-SIAM symposium on Discrete algorithms*, pages 844–853, 2008.

[27] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th annual ACM symposium on Theory of computing*, pages 604–612, 2004.

[28] R. Fadel and I. Segal. The communication cost of selfishness: ex post implementation. In *Proceedings of the 10th conference on Theoretical aspects of rationality and knowledge*, pages 165–176, 2005.

[29] N. Feamster, R. Johari, and H. Balakrishnan. Implications of autonomy for the expressiveness of policy routing. *IEEE/ACM Trans. Netw.*, 15(6):1266–1279, 2007.

[30] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[31] U. Feige. On maximizing welfare when utility functions are subadditive. In *Proceedings of the 38th annual ACM symposium on Theory of computing*, pages 41–50, 2006.

[32] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 461–471, 2007.

[33] U. Feige and J. Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 667–676, 2006.

[34] J. Feigenbaum, D. R. Karger, V. S. Mirrokni, and R. Sami. Subjective-cost policy routing. *Theor. Comput. Sci.*, 378(2):175–189, 2007.

[35] J. Feigenbaum, C. H. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. *Distributed Computing*, 18(1):61–72, 2005.

[36] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of muliticast transmissions. *Journal of Computer and System Sciences*, 63:21–41, 2001.

[37] J. Feigenbaum, V. Ramachandran, and M. Schapira. Incentive-compatible interdomain routing. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 130–139, 2006.

[38] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism design for policy routing. *Distributed Computing*, 18(4):293–305, 2006.

[39] J. Feigenbaum, M. Schapira, and S. Shenker. Distributed algorithmic mechanism design. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 14. Cambridge University Press, 2007.

[40] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: recent results and future directions. In *Proceedings of the 6th ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, 2002.

[41] D. Gale and L. Shapley. College admissions and the stability of mariage. *American Mathematical Monthly*, 69:9–15, 1962.

[42] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *Proceedings of IEEE INFOCOM*, pages 547–556, 2001.

[43] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proceedings of ACM SIGMETRICS*, pages 307–317, 2000.

[44] A. V. Goldberg, J. D. Hartline, A. R. Karlin, M. Saks, and A. Wright. Competitive auctions. *Games and Economic Behavior*, 55(2):242 – 269, 2006.

[45] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and traffic attraction: incentives for honest path announcements in BGP. In *Proceedings of ACM SIGCOMM*, pages 267–278, 2008.

[46] T. G. Griffin and G. Huston. TRFC 4264: BGP wedgies. 2005.

[47] T. G. Griffin, A. D. Jaggard, and V. Ramachandran. Design principles of policy languages for path vector protocols. In *Proceedings of ACM SIGCOMM*, pages 61–72, 2003.

[48] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10(2):232–243, 2002.

[49] T. G. Griffin and G. Wilfong. An analysis of BGP convergence properties. *SIGCOMM Comput. Commun. Rev.*, 29(4):277–288, 1999.

[50] T. Groves. Incentives in teams. *Econometrica*, pages 617–631, 1973.

[51] A. Hall, E. Nikolova, and C. H. Papadimitriou. Incentive-compatible interdomain routing with linear utilities. In *Proceedings of the Workshop on Internet Economics*, pages 232–244, 2007.

[52] S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. In *Proceedings of the 39th annual ACM symposium on Theory of computing*, pages 345–353, 2007.

[53] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[54] P. E. Haxell and G. Wilfong. A fractional model of the border gateway protocol (BGP). In *Proceedings of the 19th annual ACM-SIAM symposium on Discrete algorithms*, pages 193–199, 2008.

[55] G. Huston. Interconnection, peering, and settlements. In *Internet Global Summit (INET)*. The Internet Society, 1999.

[56] A. D. Jaggard and V. Ramachandran. Robustness of class-based path-vector systems. In *Proceedings of IEEE ICNP*, pages 84–93, 2004.

[57] H. Karloff. On the convergence time of a path-vector protocol. In *Proceedings of the 15th annual ACM-SIAM symposium on Discrete algorithms*, pages 605–614, 2004.

[58] S. Kintali. A distributed protocol for fractional stable paths problem. Technical report, Georgia Tech, College of Computing, 2008.

[59] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[60] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[61] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed internet routing convergence. *SIGCOMM Comput. Commun. Rev.*, 30(4):175–187, 2000.

[62] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 574, 2003.

[63] R. Lavi, A. Mu'alem, and N. Nisan. Two simplified proofs for Roberts' Theorem, 2004. Submitted.

[64] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behaviour*, 55(2):270–296, 2006.

[65] H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 57–66, 2008.

[66] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond moulin mechanisms. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 1–10, 2007.

[67] V. S. Mirrokni, M. Schapira, and J. Vondrak. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77, 2008.

[68] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[69] E. Mossel and C. Umans. On the complexity of approximating the VC dimension. *J. Comput. Syst. Sci.*, 65(4):660–671, 2002.

[70] H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16(2):279–320, 1999.

[71] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[72] A. Mu'alem and M. Schapira. Setting lower bounds on truthfulness. In *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, pages 1143–1152, 2007.

[73] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions ii. *Math. Programming Study 8*, pages 73–87, 1978.

[74] N. Nisan. The communication complexity of approximate set packing and covering. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 868–875, 2002.

[75] N. Nisan and A. Ronen. Computationally feasible VCG-based mechanisms. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 242–252, 2000.

[76] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 35:166 – 196, 2001.

[77] N. Nisan, M. Schapira, G. Valiant, and A. Zohar. Best-reply mechanisms. Technical report, Leibniz Center for Computer Science, Hebrew University, 2008.

[78] N. Nisan, M. Schapira, and A. Zohar. Asynchronous best-reply dynamics. In *Proceedings of the Workshop on Internet Economics*, pages 531–538, 2008.

[79] N. Nisan and I. Segal. Exponential communication inefficiency of demand queries. In *Proceedings of the 10th conference on Theoretical aspects of rationality and knowledge*, pages 158–164, 2005.

[80] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006.

[81] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 749–753, 2001.

[82] C. H. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 250–259, 2008.

[83] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *J. Comput. Syst. Sci.*, 53(2):161–170, 1996.

[84] K. Roberts. The characterization of implementable choice rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences. Papers presented at the 1st European Summer Workshop of the Econometric Society*, pages 321–349. North-Holland, 1979.

[85] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[86] T. Roughgarden. An algorithmic game theory primer. In *Proceedings of the 5th IFIP International Conference on Theoretical Computer Science. An invited survey*, 2008.

[87] T. Roughgarden and E. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236 – 259, 2002.

[88] R. Sami, M. Schapira, and A. Zohar. Security and selfishness in interdomain routing. Technical report, Leibniz Center for Computer Science, Hebrew University, Jerusalem, Israel, 2008.

[89] R. Sami, M. Schapira, and A. Zohar. Seraching for stability in interdomain routing. In *Proceedings of IEEE INFOCOM (to appear)*, 2009.

[90] N. Sauer. On the density of families of sets. *J. Comb. Theory, Ser. A*, 13(1):145–147, 1972.

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

[92] M. Schapira and Y. Singer. Inapproximability of combinatorial public projects. In *Proceedings of the Workshop on Internet Economics*, pages 351–361, 2008.

[93] I. Segal. *In P. Cramton and Y. Shoham and R. Steinberg (Editors), Combinatorial Auctions. Chapter 11. The Communication Requirements of Combinatorial Allocation Problems.* MIT Press., 2006.

[94] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific J Math*, 41:247–261, 1972.

[95] J. Shneidman and D. C. Parkes. Specification faithfulness in networks with rational nodes. In *Proceedings of the 23rd annual ACM symposium on Principles of distributed computing*, pages 88–97, 2004.

[96] J. L. Sobrinho. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking*, 13(5):1160–1173, 2005.

[97] L. Subramanian, S. Agarwal, J. Rexford, and R.H. Katz. Characterizing the internet hierarchy from multiple vantage points. *Proceedings of IEEE INFOCOM*, 2:618–627, 2002.

[98] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32(1):1–16, 2000.

[99] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, December 2007.

[100] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, pages 8–37, 1961.

[101] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 67–74, 2008.

[102] A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th annual ACM symposium on Theory of computing*, pages 209–213, 1979.