LNCS 5814

Marios Mavronicol
Vicky G. Papadopou

Algorith
Game Th

Second International Symp
Paphos, Cyprus, October 20
Proceedings

Lecture Notes in Computer Science 5814

Marios Mavronicolas Vicky G. Papadopoulou (Eds.)

Algorithmic Game Theory

Second International Symposium, SAGT 2009
Paphos, Cyprus, October 18-20, 2009
Proceedings

Springer

Volume Editors

Marios Mavronicolas
University of Cyprus
Department of Computer Science
75 Kallipoleos Str.
P.O. Box 20537
CY-1678 Nicosia, Cyprus
E-mail: mavronic@ucy.ac.cy

Vicky G. Papadopoulou
European University Cyprus
Department of Computer Science and Engineering
6 Diogenes Str.
CY-1516 Nicosia, Cyprus
E-mail: viki@ucy.ac.cy

Library of Congress Control Number: 2009934790

CR Subject Classification (1998): I.6, H.5.3, J.4, K.6.0, H.3.5, K.4.4, F.2.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-04644-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-04644-5 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12764747 06/3180 5 4 3 2 1 0

Preface

This volume contains the papers presented at the Second International Symposium on Algorithmic Game Theory (SAGT 2009), which was held on October 18–20, 2009, in Paphos, Cyprus. This event followed the first, very successful *SAGT* symposium, which took place in Paderborn, Germany, last year.

The purpose of *SAGT* is to bring together researchers from computer science, economics and mathematics to present and discuss original research at the intersection of algorithms and game theory. It has been intended to cover all important areas such as solution concepts, game classes, computation of equilibria and market equilibria, algorithmic mechanism design, automated mechanism design, convergence and learning in games, complexity classes in game theory, algorithmic aspects of fixed-point theorems, mechanisms, incentives and coalitions, cost-sharing algorithms, computational problems in economics, finance, decision theory and pricing, computational social choice, auction algorithms, price of anarchy and its relatives, representations of games and their complexity, economic aspects of distributed computing and the internet, congestion, routing and network design and formation games and game-theoretic approaches to networking problems.

Approximately 55 submissions to *SAGT 2009* were received. Each submission was reviewed by at least three Program Committee members. The Program Committee decided to accept 29 papers. Out of these, a small number will be invited to a Special Issue of the Theory of Computing Systems journal with selected papers from SAGT 2009. The program of SAGT 2009 featured three invited talks from three outstanding researchers in algorithmic game theory: Elias Koutsoupias, Dov Monderer and Mihalis Yannakakis. We are very grateful to Elias, Dov and Mihalis for joining us in Paphos and for their excellent lectures.

Our sincere thanks go to all authors who submitted their research work to SAGT 2009. We would like to thank all Program Committee members, and the external reviewers who assisted them, for their wonderful work. We are indebted to all members of the Organization Committee for their hard work preparing SAGT 2009. The developers of the EasyChair conference system, which assisted tremendously both the Program and the Organization Committees, deserve special thanks. We also thank Alan Selman, the Editor-in-Chief of the Theory of Computing Systems journal, for making the Special Issue possible.

We are very pleased to acknowledge financial support from the University of Cyprus, the Limassol Cooperative Savings Bank Ltd., IBM Cyprus and the Integrated Project AEOLUS (IST-015964) of the European Union. We are honored that SAGT is embraced under the auspices of the European Association for Theoretical Computer Science (EATCS).

October 2009

Marios Mavronicolas
Vicky G. Papadopoulou

Organization

Program Chair

Marios Mavronicolas	University of Cyprus, Cyprus

Program Committee

Petra Berenbrink	Simon Fraser University, Canada
Felix Brandt	University of Munich, Germany
Ioannis Caragiannis	University of Patras Greece
George Christodoulou	Max-Planck-Institut für Informatik Germany
Constantinos Daskalakis	MIT USA
Xiaotie Deng	City University of Hong Kong Hong Kong SAR
Federico Echenique	California Institute of Technology USA
Edith Elkind	Nanyang Technological University Singapore
Michele Flammini	University of L'Aquila Italy
Martin Gairing	University of Liverpool UK
Ron Lavi	The Technion – Israel Institute of Technology Israel
Burkhard Monien	University of Paderborn Germany
Asu Ozdaglar	MIT USA
Giuseppe Persiano	University of Salerno Italy
Tuomas Sandholm	Carnegie Mellon University USA
Maria Serna	Universitat Politecnica de Catalunya Spain
Paul Spirakis	RACTI & University of Patras Greece
Shang-Hua Teng	Boston University USA
Takashi Ui	Yokohama National University Japan
Carmine Ventre	University of Liverpool UK
Bernhard von Stengel	London School of Economics UK

Steering Committee

Elias Koutsoupias	University of Athens Greece
Marios Mavronicolas	University of Cyprus Cyprus
Dov Monderer	The Technion – Israel Institute of Technology Israel
Burkhard Monien	University of Paderborn Germany
Christos Papadimitriou	UC Berkeley USA
Giuseppe Persiano	University of Salerno Italy

Paul Spirakis	RACTI & University of Patras Greece – Chair
Berthold Vöcking	RWTH Aachen Germany

Local Organization

Chryssis Georgiou	University of Cyprus Cyprus
Christos Kapoutsis	University of Cyprus Cyprus
Marios Mavronicolas	University of Cyprus Cyprus
Vicky G. Papadopoulou	European University Cyprus Cyprus – Chair
Anna Philippou	University of Cyprus Cyprus

External Reviewers

Itai Ashlagi
Vincenzo Auletta
Umang Bhaskar
Vittorio Bilo
Patrick Briest
Markus Brill
Ning Chen
Vincent Conitzer
Nikhil Devanur
Dominic Dumrauf
Khaled Elbassioni
Angelo Fanelli
Michal Feldman
Rainer Feldmann
Diodato Ferraioli
Felix Fischer
Dimitris Fotakis
Andrew Gilpin
Paul Harrenstein
Nicole Immorlica
Albert Jiang
Panagiotis Kanellopoulos
Spyros Kontogiannis
Annamaria Kovacs
Maria Kyropoulou
Aranyak Mehta
Luca Moscardelli
Ahuva Mu'alem
Evdokia Nikolova
Abraham Othman
Mike Paterson
Paolo Penna
James Pita
Ariel Procaccia
Guido Proietti
Evangelia Pyrga
Qi Qi
Heiko Roeglin
Amir Ronen
Maher Said
Mahyar Salek
Michael Schapira
Florian Schoppmann
Ulf-Peter Schroeder
Alexander Skopalik
Troels Sorensen
Orestis Telelis
Tobias Tscheuschner
Angelina Vidali
William Walsh
Jiajin Yu
Yingchao Zhao
Martin Zinkevich
Gianpiero Monaco

Table of Contents

Monotonicity in Mechanism Design

Dov Monderer

Technion–Israel Institute of Technology
dov@ie.technion.ac.il

Abstract. Consider a model with a finite number of alternatives, and buyers with private values and quasi-linear utility functions. A domain of valuations for a buyer is a monotonicity domain if every finite-valued monotone randomized allocation rule defined on it is implementable, in the sense that there exists a randomized truth-telling direct mechanism, which implements this allocation rule. The domain is a weak monotonicity domain if every deterministic monotone allocation rule defined on it is implementable. I discuss the literature on (weak) monotonicity domain, which includes the early mathematical literature as well as the recent CS/Economics literature.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, p. 1, 2009.

Computational Aspects of Equilibria

Mihalis Yannakakis

Department of Computer Science, Columbia University

1 Introduction

Equilibria play a central role in game theory and economics. They characterize the possible outcomes in the interaction of rational, optimizing agents: In a game between rational players that want to optimize their payoffs, the only solutions in which no player has any incentive to switch his strategy are the Nash equilibria. Price equilibria in markets give the prices that allow the market to clear (demand matches supply) while the traders optimize their preferences (utilities). Fundamental theorems of Nash [34] and Arrow-Debreu [2] established the existence of the respective equilibria (under suitable conditions in the market case). The proofs in both cases use a fixed point theorem (relying ultimately on a compactness argument), and are non-constructive, i.e., do not yield an algorithm for constructing an equilibrium. We would clearly like to compute these predicted outcomes. This has led to extensive research since the 60's in the game theory and mathematical economics literature, with the development of several methods for computation of equilibria, and more generally fixed points. More recently, equilibria problems have been studied intensively in the computer science community, from the point of view of modern computation theory. While we still do not know definitely whether equilibria can be computed in general efficiently or not, these investigations have led to a better understanding of the computational complexity of equilibria, the various issues involved, and the relationship with other open problems in computation. In this talk we will discuss some of these aspects and our current understanding of the relevant problems. We outline below the main points and explain some of the related issues.

2 General Setting

Consider a game G in normal form, i.e., given by the payoff tables of the players. As usual in computer science, all input data (the payoffs in this case) are assumed to be rational. However this does not mean that the equilibria will be also rational: As was observed already by Nash, there are even 3-player games whose equilibria are irrational. This is a common phenomenon in many areas in science and engineering, where the quantities of interest are described by nonlinear equations. The usual approach is to compute a rational approximation to the desired quantities within a desired precision (error) ϵ; for example, π is approximately 3.14159 up to five decimal digits of precision, i.e. within error $\epsilon < 10^{-5}$. Given a game G, we would like to compute a Nash equilibrium for it, exactly if possible, or approximately within a given precision ϵ.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 2–13, 2009.

The complexity of the Nash equilibrium problem is intimately connected with the complexity of the computation of fixed points. As mentioned above, Nash proved the existence of an equilibrium, by constructing from a game G a continuous function F_G from the set D of all real-valued vectors that represent mixed strategy profiles of the players to itself, such that the set of fixed points of F_G is precisely the set of equilbria of G; since D is a convex compact domain and F_G is continuous, there is at least one fixed point by Brouwer's theorem. Nash's function F_G is specified by a simple algebraic formula using the operators $+, -, *, /, \max$. Thus, the computation of a Nash equilibrium for a given game (exactly, or approximately within given precision) reduces to the (exact or approximate) computation of a fixed point for a specific algebraic Brouwer function.

It turns out that a strong converse is also true: Nash equilibria are powerful enough to capture the fixed points of *all* algebraic Brouwer functions. Let FIXP be the class of total search problems that can be formulated as fixed point computation problems of algebraic Brouwer functions: a problem Π is in FIXP if from a given instance I we can construct in polynomial time a circuit (equivalently, a straight-line program) over basis $\{+, -, *, /, \max\}$ with rational constants that computes a continuous function F_I from a domain D_I to itself, such that the solutions of the instance I are the fixed points of F_I, i.e. points $x \in D_I$ that satisfy $x = F_I(x)$; the domain could be a polyhedral domain (e.g., simplex, hypercube, or a more complex polytope) or a nonlinear domain such as an ellipsoid. The Nash equilibrium problem obviously belongs to FIXP by Nash's proof. Furthermore it is FIXP-complete for games with 3 or more players [15]. This holds with respect to both exact and approximate computation. From an instance I of a problem $\Pi \in$ FIXP we can construct in polynomial time a 3-player game G whose Nash equilibria yield the solutions to the instance I by a simple linear transformation with rational coefficients. Similarly with approximate computation: To compute a solution of I within a desired number k of bits of precision (i.e. error $\epsilon = 2^{-k}$) it suffices to compute an equilibrium of G within k' bits of precision, where k' is polynomial in I and k.

The class FIXP is quite robust, both with respect to the choice of the domain as mentioned above, as well as with respect to the representation of the function and the basis set of operators. Thus, for example the class remains the same whether we use formulas or circuits to represent the function. Also it can be shown that the class does not change if we add roots and fractional powers to the basis, nor if we remove division. Many problems from various areas can be formulated as fixed point computation problems for algebraic functions; see [15] for several examples. In particular, computing a price equilibrium in the classical exchange economy market model where the demand functions of the traders are given explicitly by algebraic formulas (or circuits) is in FIXP by the standard fixed point formulation of the market equilibrium problem; furthermore, the problem is also FIXP-complete.

The various discrete computational problems associated with problems in FIXP, can be solved in polynomial space; this includes the computation of a solution (fixed point) within a desired precision, as well as more complex *existence questions*, such as whether there exists a solution that satisfies given conditions (for example, has certain coordinates in a specified range). All these problems can be expressed in the existential theory of the reals, via the fixed point equations, and solved in PSPACE using a procedure for this theory. Existence questions are generally NP-hard; for example, it is NP-hard to tell whether a given 2-player game has a Nash equilibrium whose support includes specified strategies, or an equilibrium in which a player receives a specified payoff [22]. However, this does not mean that it is NP-hard to compute (approximately) an equilibrium (anyone) of a game; this is an open question. As far as we currently know, FIXP is somewhere between P and PSPACE.

Although there is no known NP-hardness lower bound for the exact or approximate computation of equilibria, the complexity of the equilibria problems are lower bounded by other longstanding open problems in computation. One such problem is the square-root-sum (Sqrt-Sum) problem, of determining whether the sum of square roots of a given set of integers is greater than another integer; this problem arises often especially in geometric computations, and it is a 30-year old question whether it is in NP (or even in P). A second, and more fundamental, problem, called PosSLP, is to determine whether a given circuit (or Straight-Line-Program) with given inputs 0, 1 and operations $+, -, *$ computes a positive number. The significance of this problem is that it characterizes the power of the unit-cost model of computation with unbounded precision rational arithmetic, i.e., a model in which all operations $(+, -, *, /)$ on rational numbers take unit time regardless of the size of the numbers [1]; this model is known also as the algebraic RAM model, and corresponds essentially to the real computation model of [5] restricted to rational constants. The Sqrt-Sum problem can be solved in P-time in this model [47]. It is known that if we have integer division (even just division by 2, i.e., the floor operation $\lfloor \frac{x}{2} \rfloor$), then all of PSPACE can be done in polynomial time in the unit cost model [4]. The unit cost rational model appears to be somewhat weaker than PSPACE: it was shown recently that P-time in this model is contained in the Counting Hierarchy, a hierarchy above #P and the Polynomial Hierarchy [1]. The model appears still to be quite powerful and its relationship to the basic complexity classes in the standard Turing model (P, NP) remains open.

The Sqrt-Sum and PosSLP problems can be reduced to (and thus lower bound) the problem of computing approximately with any nontrivial accuracy a Nash equilibrium of a given 3-player game, or the prices in an equilibrium of an exchange economy [15]. This lower bound holds even for games and markets that are guaranteed to have a unique equilibrium. The issue of uniqueness and the problem of equilibrium selection in games and markets that have multiple equilibria is an important issue that has received a lot of attention. As in many other fields, it is desirable for a theory to predict a unique outcome; see e.g. [23] and the remarks of Aumann regarding the role of uniqueness in games, and [33], Chapter 17 regarding market equilibria. Of course, if there is only one equilibrium to begin with, then there is no ambiguity: the unique equilibrium is the predicted outcome of the game or the market, and the problem is thus to compute or estimate the values (strategy probabilities or prices) in the unique equilibrium. This problem seems to be hard. Specifically, given a game that has a unique Nash equilibrium, or given a market (with explicit excess demand functions) that has a unique price equilibrium, if we can distinguish in polynomial time whether the probability of a particular strategy in the (unique) equilibrium of the given game or the price of a particular good in the given market is very close to 0 from the case that it is very close to 1, then the unit cost rational model can be essentially simulated in polynomial time in the standard Turing model; i.e., unit-cost-P $\subseteq$ P. This is very unlikely.

3 Two-Player Games and Approximate Equilibria

The case of 2-player games has certain nice properties that do not hold for 3 or more players. One such property is that, if the payoffs are rational, then there is always a rational Nash equilibrium with polynomial bit complexity. Furthermore, if we know the support of an equilibrium (and there is a finite, though exponential, number of possible supports), then we can compute explicitly an equilibrium with this support by setting up and solving a system of linear equations. This means in particular that the problem of computing an equilibrium is in NP, since we can guess a support and compute and verify a corresponding equilibrium. The problem cannot be NP-hard unless NP=coNP (which is widely believed not to be the case). In the special case of *zero-sum* 2-player games (i.e., if the two players get opposite payoffs for every strategy pair), an equilibrium can be computed in P by Linear Programming.

In the general (non-zero sum) 2-player case, the equilibrium problem corresponds to a Linear Complementarity problem. An equilibrium can be computed by the Lemke-Howson algorithm [31]; the algorithm runs in exponential time in the worst case [40], but this of course does not rule out the existence of other, more efficient algorithms. Papadimitriou defined in [37] a complexity class PPAD (motivated primarily by this problem) which captures the basic path-following principle of the Lemke-Howson algorithm and the similar algorithm of Scarf for computing approximate fixed points [41]. A recent sequence of papers culminated in showing that computing a Nash equilibrium in 2-player games is PPAD-complete [11,7]. Furthermore, it is also complete to compute an approximate ϵ-Nash equilibrium (i.e. a (mixed) strategy profile in which no player can improve its payoff by more than ϵ by switching strategy unilaterally), even if all the payoffs are 0 or 1 and ϵ is inverse polynomial in the number of strategies [8]. If all the payoffs are in $[0, 1]$ (or have a bounded range) and ϵ is constant, then a ϵ-Nash equilibrium can be computed in quasipolynomial time, specifically in time $n^{O(\log n/\epsilon^2)}$, for any fixed number of players [32]. It is open whether there is a polynomial time (additive) approximation scheme, even for 2 players. There are several papers that give polynomial-time algorithms for some constant ϵ for 2-player games with payoffs in $[0, 1]$; see [46] for an overview of the progress so far on the constant ϵ.

The class PPAD is somewhere between P and NP, more precisely, TFNP, the class of total search problems in NP. As with all total NP problems, the problems in PPAD cannot be NP-hard unless NP=coNP. A characterization of PPAD in terms of fixed points is that PPAD is the class of search problems that can be expressed as fixed point computation problems for polynomial piecewise linear functions [15]. PPAD is contained in FIXP, and more precisely, it corresponds to the piecewise linear fragment of FIXP, which is obtained by restricting the circuits in the definition of FIXP to allow multiplication and division only with rational constants, and not between any gates and inputs variables. The functions computed by such circuits are piecewise linear, and they always have rational fixed points.

The problem of computing an ϵ-Nash equilibrium (ϵ-NE) for a given game G and (rational) $\epsilon > 0$ is in PPAD (and PPAD-complete) for any number of players, and the same holds for the slightly stronger notion of *ϵ-well supported Nash equilibrium* (ϵ-ws NE) [11]. A ϵ-ws NE is an ϵ-Nash equilibrium with the additional property that pure strategies whose payoff is suboptimal by more than ϵ have 0 probability. These notions are natural relaxations of the equilibrium solution concept, allowing more solutions where the players' incentive to switch strategy is small. The two versions are polynomially related: every ϵ-ws NE is by definition an ϵ-NE, and moreover for any game G and ϵ, we can take ϵ' of bit-size (i.e. $\log(1/\epsilon')$) polynomial in G and the bit-size of ϵ, such that every ϵ'-NE of G is an ϵ-ws NE [11]. The well-supported version (but not plain ϵ-NE) has furthermore the following property characterizing exact equilibria of nearby games: A mixed strategy profile x is a ϵ-well supported Nash equilibrium of a game G if and only if there is a game G' that differs from G at most by $\epsilon/2$ in each payoff such that x is a (exact) equilibrium of G' [19].

4 Relationship between Types of Approximation

Approximate Nash equilibria of a game G (in both the ϵ-NE and ϵ-ws NE versions) are quantitatively different than approximations of (actual) Nash equilibria of G: the fact that a strategy profile is almost at equilibrium, in the sense that the incentive to move is small, does not necessarily mean that it is near an equilibrium. (This phenomenon is familiar also in the physical world: the net force in a configuration may be small, but the system may still be far from equilibrium.) If a strategy profile is near an equilibrium then it is easy to see that it is almost at equilibrium. But the converse does not hold. In particular, there are examples of 3-player games G of size n with a unique Nash equilibrium such that G has ϵ-NE (and ϵ-ws NE) for an extremely small $\epsilon = 1/2^{2^{n^c}}$, that are almost at distance 1 from the equilibrium, i.e., some strategy probability is close to 1 in the ϵ-NE whereas it is close to 0 in the (unique) equilibrium [15]. This phenomenon is due to the nonlinearity of the Nash equilibrium problem from 3 players on; in particular, it does not happen for 2-player games.

More generally, similar issues arise in the relationship between ϵ-approximation to fixed points of a function F on the one hand and ϵ-approximate fixed points of

F on the other hand, i.e., points x such that $|F(x) - x| \leq \epsilon$; these are referred respectively as *strong* and *weak* approximation because the second one polynomially reduces to the first one for most common functions (technically, for polynomially continuous functions). The converse reduction generally does not hold. In the case of game equilibria, ϵ-NE correspond to (are polynomially related to) the weak approximation for Nash's function.

If we do not care about the complexity of the problems, then the strong and the weak version are related asymptotically to each other as they both converge to fixed points as $\epsilon \to 0$, so it is perhaps not so important asymptotically to make a big distinction between them. Specifically, the following qualitative converse holds: for every Brouwer function F, for every $\epsilon > 0$, there is a $\delta > 0$ such that every point x that satisfies $|F(x) - x| \leq \delta$ is within ϵ of some fixed point of F [29]; for a class of smooth functions, δ is moreover linear in ϵ [3]. These results indicate that the problem of approximating a fixed point (e.g., an equilibrium) reduces to the problem of computing an approximate fixed point; it means for example that we can use algorithms for the computation of approximate fixed points based on simplicial subdivision, such as those of Scarf, Kuhn, McKennon and others [41,29] to actually compute approximations of fixed points, by using a fine enough subdivision. The problem is that the reduction is not a polynomial reduction: the value of δ that we need depends not only on ϵ but also on the function F, i.e., on the instance of the problem in hand. For example, in the case of the Nash equilibrium problem, F is Nash's function for the given game, and we know that δ has to be at least doubly exponentially small in magnitude (i.e. of exponential bit-size) in the size of the game.

If we consider the complexity of the problems, it is essential to distinguish carefully between the two types of approximation: *computing approximately an equilibrium* (or even exactly if possible, e.g. if it is rational) on the one hand versus *computing an approximate equilibrium* on the other hand. Failing to make the distinction can lead to wrong conclusions: We know that the latter problem of computing an approximate equilibrium is in PPAD, and thus in NP. However, if the former problem of computing approximately an equilibrium is in PPAD, this would have immediate consequences on longstanding open problems; it would imply in particular that the Sqrt-Sum and posSLP problems are in NP, and that P-time in the unit-cost rational (algebraic RAM) model is contained in NP in the standard Turing model. We do not know currently the status of these problems, and the last statement is in fact very doubtful. We believe actually that approximate computation of equilibria is most likely not in NP.

The difference between the two types of approximations is manifested starkly in a setting where the algorithm is not given explicitly full information about the instance at hand (i.e., the specific function in the case of a fixed point problem, the strategies and payoffs of the players in a game, the utilities or demand functions of the traders in a market), but rather information is obtained dynamically as the algorithm progresses. In these cases, the limitations on the access of the algorithm to the instance permit one to show unconditional bounds.

5 Price Adjustment Mechanisms

The question of how markets arrive at equilibrium prices that match demand and supply has occupied economists for a very long time, starting with Walras' proposed tatonnement mechanism more than a century ago. Although tatonnement does not converge to an equilibrium in certain markets (as shown by Scarf), several other price adjustment mechanisms have been proposed that do converge to an equilibrium [45,30,24,25]. The mechanisms adjust dynamically the prices according to the excess demands of the goods. The demand functions are assumed typically to be differentiable, and the adjustment of prices is described mathematically by differential equations based on the demands and their derivatives. Adjustment mechanisms that converge to equilibria have been also proposed for games, and serve also a role of selecting an equilibrium when there are multiple equilibria [23]; see [25] for an overview. Price adjustment mechanisms can be seen as algorithms that compute equilibria, but they do not use explicitly the excess demand functions directly, but only indirectly through the observed values (and derivatives).

Consider a general discrete-time price adjustment scheme, the prices are updated in every step based on the excess demands at the current prices as well as the whole history of prices and demands (and derivatives of demands). Assume we restrict to markets that have continuously differentiable excess demand functions with bounded derivatives and which have a unique equilibrium. The time required for a price-adjustment mechanism to arrive at an ϵ-approximate equilibrium, i.e., a set of prices for which the excess demand of each good is at most an ϵ fraction of the total supply, is exponential in the number of goods (where the base of the exponent depends on ϵ); the time to converge within ϵ of the equilibrium is not bounded by any function of ϵ (even for three goods) [38].

Technically, these results are shown by extending to differentiable functions and the price simplex analogous lower bounds of [26,44] on the computation of a fixed point of a (unknown) Brouwer function F on a hypercube using a generic black-box algorithm that can access the function only by evaluating it at individual points. The time (number of function evaluations) required for a black-box algorithm to compute a weak ϵ-approximate fixed point is exponential in the dimension, and the time required to compute a strong ϵ-approximate fixed point is unbounded (even for dimension 2).

6 Dynamic, Stochastic Games

The previous discussion on the complexity of Nash equilibria concerned one-shot games given in normal form, i.e. all the (pure) strategies are listed explicitly, and the payoffs of the players are given for all possible combinations of pure strategies. In many situations, the games are specified implicitly and they take place dynamically over time; as a result, the number of pure strategies is much larger than the specification, exponential or even infinite. We will discuss here briefly dynamic, stochastic games and the complexity of their equilibria problems.

A typical dynamic game Γ has a set V of states, and every state u has an associated (one-shot) game G_u. The game Γ starts at some state and proceeds in discrete steps from one state to another. In each step, if the game is at state u, the players choose their actions (pure strategies) in the corresponding game G_u, based possibly on the whole past history and possibly using randomization, and then the play proceeds to a new state v with probability that depends on the actions selected by the players. The objectives of the players fall into two main types: one is based on rewards (payoff) accumulated during the execution, and the other type is based on a property of the whole execution. In the first type of objective, the specification of the game G_u includes a reward to each player for each combination of actions, as in a standard one-shot normal form game; the rewards r_t of a player in the different steps $t = 1, 2, \ldots$ of the execution are aggregated using an aggregation function, such as discounted sum, $\sum_{t=1}^{\infty} \lambda^t r_t$ (where $\lambda < 1$ is the discount factor), or long-term average, e.g. $\limsup_{n\to\infty}(\sum_{t=1}^{n} r_t)/n$, and the player wants to maximize the aggregate reward. In the second type of objective, a player wants to maximize the probability that the execution satisfies a certain property, such as a 'reachability' property (the execution visits a state in a specified set S), or its opposite, called a 'safety' property (no state of S is visited), or an 'infinite reachability' property (a state in S is visited infinitely often), etc. For computational purposes, it is assumed as usual that all input data (transition probabilities, rewards) are rational.

Assume the state set V is finite. If there is only one player and probabilistic transitions, then Γ is a Markov Decision Process, a model which has a well developed theory and efficient algorithms for a variety of problems, see e.g. [21]. Already for two-player, zero-sum games, the problems become quite challenging and there are many important open questions for various classes of such games. The (2-player 0-sum) games have generally a well-defined *value*; the problem is to compute the value for a given game and optimal strategies for the players. We can classify the games into two types: *turn-based* games which have the restriction that at each state only one player has a choice of action, and *concurent* games, in which there is no such restriction, i.e., both players can have choices at the same state. Turn-based games have typically several nice properties: the values are rational, and the players have pure (deterministic) memoryless optimal strategies (i.e. they can always pick the same single action at each node) that achieve the value. In concurrent games the value is generally irrational, there may exist only ϵ-optimal strategies and they may require randomization.

Shapley's original stochastic games [43] are 2-player 0-sum concurrent games with a discounted aggregate reward objective. The value v is in general irrational, and we do not know of any efficient way to approximate it or to answer a decision question such as $v > 0$? (e.g., should player 1 want to play the game?). The problem of computing the value of the game is in FIXP. The decision question is at least as hard as the Sqrt-Sum problem, thus we do not know if it is in NP. The approximation problem on the other hand is in PPAD [15].

A simpler game, called *simple stochastic game* (SSG) was introduced by Condon [10]. This is a 2-player, 0-sum game with a reachability objective for player 1 (i.e. he wants to reach a state in a set S), and the complementary safety objective for player 2 (i.e. he wants to avoid the state set S). The value v is rational of polynomial bit complexity, but again we do not know of any efficient way to compute it, or approximate it, or answer a decision question such as $v > 1/2$?. We can determine efficiently whether $v = 0$ or $v = 1$, but comparing with any other constant in-between is open. The decision question is in NP∩coNP [10] (and in fact in UP∩coUP). The problem of computing the value is in PPAD [15] as well as in the class PLS [49] of Polynomial Local Search problems. PLS is a class of total search problems between P and NP introduced in [27]. Typical problems include finding locally optimal solutions to combinatorial problems and computing pure equilibria of games such as congestion games where they are guaranteed to exist [20]; see [50] for more information on PLS. We note incidentally that for both Condon's and Shapley's games, the values of the game for various starting states can be expressed as the unique solution to a fixed point set of equations $x = F(x)$; in both cases it is easy to find a weak ϵ-approximate fixed point (i.e., point x such that $|F(x) - x| \leq \epsilon$) for ϵ inverse polynomial, however such a point gives no information on the desired values of the game.

Two even simpler classes of turn-based 2-player 0-sum non-stochastic games are the *mean payoff games* [13] which are games with an average reward objective, and *parity games* [14], which have a property-based objective and play a prominent role in the verification area. Both classes reduce to simple stochastic games [52,28,39]. Their decision problem for the value of the game is in both cases in NP∩coNP, and it is open whether it is in P.

In the other direction, simple and concurrent stochastic games can be extended with a recursive feature to model recursive programs and systems that combine probabilistic and controlled aspects [18,17,16]. We will not give the detailed definitions here. A 1-RSSG (Recursive Simple Stochastic Game) essentially consists of several finite-state component SSGs, each component has a designated entry state where it starts and a designated exit state where it terminates, and the components can call each other recursively like recursive procedures. A 1-RSSG is a compact representation of an infinite-state simple stochastic game. The objective of Player 1 is to terminate the game, i.e. reach the exit state of the component of the starting state, and Player 2 has the opposite objective. The value is generally irrational in this case (this is so even if there are no players, i.e. for Recursive Markov Chains). The decision problem (e.g., is the value $v > 1/2$?) in this case is Sqrt-Sum-hard, and thus, it is not known to be in NP. The qualitative problem of determining whether the value is 1 (i.e. whether Player 1 can achieve termination almost surely) is in NP∩coNP, and is at least as hard as the (quantitative) decision problem for Condon's finite-state simple stochastic games [17]. In the case of the analogous concurrent 1-RCSG model with a termination objective, even the qualitative value=1 problem is Sqrt-Sum-hard [18].

7 Conclusions

We outlined some of the main aspects in the computation of equilibria, and discussed some of the related issues. A more detailed overview of equilibria, fixed points, and the complexity classes that capture them can be found in [51]. Although there has been significant progress in our understanding of the algorithmic problems involved in the computation of equilibria, a lot of open questions remain, both in the general setting, as well as for many interesting and important classes of games.

Acknowledgement. Work partially supported by NSF Grant CCF-0728736.

References

1. Allender, E., Bürgisser, P., Kjeldgaard-Pedersen, J., Miltersen, P.B.: On the complexity of numerical analysis. SIAM J. Computing 38(5), 1987–2006 (2009); Preliminary version in Proc. 21st IEEE Comp. Compl. Conf. (2006)
2. Arrow, K.J., Debreu, G.: Existence of an equilibrium for a competitive economy. Econometrica 22, 265–290 (1954)
3. Anderson, R.M.: "Almost" implies "Near". Trans. Am. Math. Soc. 296, 229–237 (1986)
4. Bertoni, A., Mauri, G., Sabadini, N.: A characterization of the class of functions computable in polynomial time on random access machines. Proc. ACM Symp. Th. of Comp., 168–176 (1981)
5. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, Heidelberg (1998)
6. Chaterjee, K., Doyen, L., Henzinger, T.: A survey of stochastic games with limsup and liminf objectives. In: Proc. 36th ICALP, Part II, pp. 1–15 (2009)
7. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. J. of ACM 56(3) (2009); Preliminary version in FOCS 2006
8. Chen, X., Teng, S.H., Valiant, P.: The approximation complexity of win-lose games. In: Proc. 18th ACM SODA (2007)
9. Cole, R., Fleischer, L.: Fast-converging tatonnement algorithms for one-time and ongoing market problems. In: Proc. 40th ACM STOC (2008)
10. Condon, A.: The complexity of stochastic games. Inf. & Comp. 96(2), 203–224 (1992)
11. Daskalakis, C., Goldberg, P., Papadimitriou, C.: The complexity of computing a Nash equilibrium. SIAM J. on Computing (2009); Preliminary version in STOC 2006
12. de Alfaro, L., Henzinger, T.A., Kupferman, O.: Concurrent reachability games. In: Proc. IEEE FOCS, pp. 564–575 (1998)
13. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Intl. J. Game Theory 8, 109–113 (1979)
14. Emerson, E.A., Jutla, C.: Tree automata, μ-calculus and determinacy. In: Proc. IEEE FOCS, pp. 368–377 (1991)
15. Etessami, K., Yannakakis, M.: On the complexity of Nash equilibria and other fixed points. Proc. IEEE FOCS (2007)

16. Etessami, K., Yannakakis, M.: Recursive Markov decision processes and recursive stochastic games. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 891–903. Springer, Heidelberg (2005)
17. Etessami, K., Yannakakis, M.: Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 634–645. Springer, Heidelberg (2006)
18. Etessami, K., Yannakakis, M.: Recursive concurrent stochastic games. Logical Methods in Computer Science 4(4:7), 1–21 (2008); Preliminary version in: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 324–335. Springer, Heidelberg (2006)
19. Etessami, K., Yannakakis, M.: Manuscript (2009)
20. Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure Nash equilibria. In: Proc. ACM STOC, pp. 604–612 (2004)
21. Filar, J., Vrieze, K.: Competitive Markov Decision Processes. Springer, Heidelberg (1997)
22. Gilboa, I., Zemel, E.: Nash and correlated equilibria: some complexity considerations. Games and Economic Behavior 1, 80–93 (1989)
23. Harsanyi, J.C., Selten, R.: A General Theory of Equilibrium Selection in Games. MIT Press, Cambridge (1988)
24. Herings, P.J.-J.: A globally and universally stable price adjustment process. J. of Mathematical Economics 27(2), 163–193 (1997)
25. Herings, P.J.-J.: Universally converging adjustment processes - a unifying approach. J. of Mathematical Economics 38, 341–370 (2002)
26. Hirsch, M.D., Papadimitriou, C.H., Vavasis, S.A.: Exponential lower bounds for finding Brower fixed points. J. Complexity 5, 379–416 (1989)
27. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? J. Comp. Sys. Sci. 37, 79–100 (1988)
28. Jurdzinski, M.: Deciding the winner in parity games is in UP∩coUP. Inf. Proc. Let. 68, 119–124 (1998)
29. Kuhn, H.W., MacKinnon, J.G.: Sandwich methods for finding fixed points. J. Opt. Th. Appl. 17, 189–204 (1975)
30. van der Laan, G., Talman, A.J.J.: A convergent price adjustment process. Economics Letters 23, 119–123 (1987)
31. Lemke, C., Howson, J.: Equilibrium points of bimatrix games. J. SIAM, 413–423 (1964)
32. Lipton, R.J., Markakis, E., Mehta, A.: Playing large games using simple strategies. In: Proc. ACM Conf. Elec. Comm, pp. 36–41 (2003)
33. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford Univ. Press, Oxford (1995)
34. Nash, J.: Non-cooperative games. Annals of Mathematics 54, 289–295 (1951)
35. Neyman, A., Sorin, S. (eds.): Stochastic Games and Applications. Kluwer, Dordrecht (2003)
36. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge Univ. Press, Cambridge (2007)
37. Papadimitriou, C.: On the complexity of the parity argument and other inefficient proofs of existence. J. Comput. Syst. Sci. 48(3), 498–532 (1994)
38. Papadimitriou, C., Yannakakis, M.: An impossibility theorem for price adjustment mechanisms, manuscript (2009)

39. Puri, A.: Theory of hybrid systems and discrete event systems. PhD Thesis, UC Berkeley (1995)
40. Savani, R., von Stengel, B.: Hard to solve bimatrix games. Econometrica 74, 397–429 (2006)
41. Scarf, H.: The approximation of fixed points of a continuous mapping. SIAM J. Appl. Math. 15, 1328–1343 (1967)
42. Scarf, H.: The Computation of Economic Equilibria. Yale University Press (1973)
43. Shapley, L.S.: Stochastic games. Proc. Nat. Acad. Sci. 39, 1095–1100 (1953)
44. Sikorski, K.: Optimal solution of nonlinear equations. Oxford Univ. Press, Oxford (2001)
45. Smale, S.: A convergent process of price adjustment and global Newton methods. J. of Mathematical Economics 3(2), 107–120 (1976)
46. Spirakis, P.: Approximate equilibria for strategic two-person games. In: Proc. 1st Symp. Alg. Game Theory, pp. 5–21 (2008)
47. Tiwari, P.: A problem that is easier to solve on the unit-cost algebraic RAM. J. of Complexity, 393–397 (1992)
48. Uzawa, H.: Walras' existence theorem and Brower's fixpoint theorem. Econ. Stud. Quart. 13, 59–62 (1962)
49. Yannakakis, M.: The analysis of local search problems and their heuristics. In: Proc. STACS, pp. 298–311 (1990)
50. Yannakakis, M.: Computational complexity of local search. In: Aarts, E.H.L., Lenstra, J.K. (eds.) Local Search in Combinatorial Optimization. John Wiley, Chichester (1997)
51. Yannakakis, M.: Equilibria, fixed points, and complexity classes. Computeer Science Review 3(2), 71–86 (2009); Preliminary version in STACS 2008
52. Zwick, U., Paterson, M.S.: The complexity of mean payoff games on graphs. Theoretical Computer Science 158, 343–359 (1996)

A Modular Approach to Roberts' Theorem*

Shahar Dobzinski[1] and Noam Nisan[1,2,**]

[1] The School of Computer Science and Engineering,
The Hebrew University of Jerusalem
shahard@cs.huji.ac.il
[2] Google Tel Aviv
noam@cs.huji.ac.il

Abstract. Roberts' theorem from 1979 states that the only incentive compatible mechanisms over a full domain and range of at least 3 are weighted variants of the VCG mechanism termed affine maximizers. Roberts' proof is somewhat "magical" and we provide a new "modular" proof. We hope that this proof will help in future efforts to extend the theorem to non-full domains such as combinatorial auctions or scheduling.

1 Introduction

Mechanism design theory has gained a place as a conceptual cornerstone for designing computer protocols among self-interested parties, as is found in the internet. For background on mechanism design we refer the reader to standard textbooks in micro-economic theory [10] and for background on its computational applications to part II of [13].

The most basic notion in mechanism design is that of truthfulness in dominant strategies. The setting involves a set of alternatives A, and a set of n players, that each has a valuation function $v_i \in V_i \subseteq \Re^A$, where $v_i(a)$ denotes the value that player i assigns to alternative a. A (direct revelation) mechanism $M = (f, p_1, ..., p_n)$ contains a preference-aggregation function $f : V_1 \times \cdots \times V_n \to A$ and payment functions $p_i : V_1 \times \cdots \times V_n \to \Re$. An incentive compatible mechanism ensures that each player's best interest when "reporting" his input v_i to the mechanism is to report it truthfully.

Definition 1. *A mechanism $M = (f, p_1, ..., p_n)$ is incentive compatible (equivalently truthful or strategy-proof) if for all players i, all valuations $v_i, v_i' \in V_i$ and $v_{-i} \in V_{-i}$,*

$$v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v_i', v_{-i})) - p_i(v_i', v_{-i})$$

We say that f can be implemented *if for some $p_1, ..., p_n$, the obtained mechanism is incentive compatible.*

* Supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities, and by a grant from the Israel Academy of Sciences.

** Corresponding author.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 14–23, 2009.

As incentive compatibility is the basic requirement in applications, a characterization is of central interest:

Characterization Question. Which preference-aggregation functions f are implementable?

While it is clear why we would like to understand which naturally-desired functions are implementable, in computational applications we require more: understanding the implementability of the family of approximations to a desired function. The necessity of settling for an approximation can be either due to computational hardness of f or due to the unimplementability of f itself.

The key positive result in mechanism design, VCG mechanisms (see, e.g., [12]), states that social-welfare maximization is implementable for every range of alternatives and domain of valuations. I.e., the function $f(v_1, ..., v_n) = \arg\max_a \sum_i v_i(a)$ is implementable with VCG payments. (In the case that maximum social welfare is obtained at more than alternative, any of them can be chosen.) It is easy to generalize the VCG mechanisms to weighted variations, termed affine maximizers:

Definition 2. *$f : V_1 \times \cdots \times V_n \to A$ is an* affine maximizer *if there exist real constants $\alpha_1, ..., \alpha_n$, $\alpha_i \geq 0$, and β_a for all $a \in A$ such that for all v, $f(v) \in \arg\max_a \sum_i(\alpha_i v_i(a)) + \beta_a$.*

The main impossibility result in the area is the surprising theorem of Roberts:

Theorem 1 ([15]). *: The only implementable functions with a finite range of size more than 2, $|A| \geq 3$ and full domains $V_i = \Re^A$ are affine maximizers.*

The main restriction in the theorem is the requirement of full domain. The extreme opposite case is where the domain is essentially single dimensional. In these cases, termed "single-parameter", much more can be implemented, and a good characterization is known (see, e.g., [12]), which has been used extensively in computational settings [1,9,4]. The case of $|A| = 2$ is always single parameter, which explains why the theorem required $|A| \geq 3$.

Most interesting computational applications do not have a full domain nor are they single-parameter, and indeed we do not have good characterizations of which functions are implementable for, e.g., combinatorial auctions, multi-unit auctions, or scheduling. All of the few such characterizations known are in very restricted settings: restricted auction domains [7,6] and "combinatorial public projects" [14]. All these papers start by showing that all implementable mechanisms are affine maximizers by proving Roberts-like theorems, and then provide lower bounds on what can be achieved by affine-maximizers in polynomial time [5]. This lack of characterization is the underlying reason for the very little progress on the long standing problems of how well can computationally-efficient incentive compatible mechanisms approximate the optimal allocation in combinatorial auctions, or approximate the optimal schedule in scheduling problems. Of note is a positive result for "combinatorial auctions with duplicate items" [3].

Extending Roberts' theorem to other domains has remained elusive. While Roberts' proof itself is not very difficult or long, it is quite mysterious (to us, at least). There is no clear separation into independent tasks, each which can be extended (or not) to non-full domains. The second author has already been involved in efforts to extend [7] and simplify the proof of [8] Roberts' theorem, but still finds it mysterious. During the last few years the two authors have spend considerable time in attempting to extend — or at least obtain a really clear proof of — Roberts' theorem. While we can not claim to be completely satisfied, we feel that we do have a new modular proof that may be of interest and so we bring it here.

Our Approach. The proof starts by considering the case where there are only two players. The first novel step is to show that there exists a player with no veto power. I.e., that for every value of v_i, the range of f after fixing this value still remains full. Having this property, we are able to show that the only implementable mechanisms for two players are affine maximizers. Then, the proof proceeds by induction on the number of players, showing that if all truthful mechanisms for $n-1$ players are affine maximizers, then all truthful mechanisms for n players must be affine maximizers too. In a sense the last step shows that as far as characterizations are concerned, we can restrict our attention to the "simpler" setting of only 2 bidders.

From a technical point of view, our proof is completely combinatorial and does not rely on the separation theorem between convex bodies, unlike the original proof. Here is a high-level structure of the new proof.

1. We begin with the direct and standard [12] observations:
   - An implementable f is "weakly monotone": $f(v_i, v_{-i}) = a$ and $f(v'_i, v_{-i}) = b$ implies that $v_i(a) - v_i(b) \geq v'_i(a) - v'_i(b)$.
   - The payment function for player i does not depend on v_i, and may be represented as $p_i(v_i, v_{-i}) = p_i^a(v_{-i})$ for $a = f(v_i, v_{-i})$.
   - f must optimize for each player: $f(v_i, v_{-i}) \in \arg\max_a v_i(a) - p^a(v_{-i})$.
2. The next step is due to [7] and shows that "ties can be ignored". Specifically, we may assume without loss of generality that the "$\geq$" in the weak monotonicity condition is in fact strict, a condition termed strong monotonicity: $f(v_i, v_{-i}) = a$ and $f(v'_i, v_{-i}) = b$ implies that $v_i(a) - v_i(b) > v'_i(a) - v'_i(b)$. As shown in [7] a the critical element here is full-dimensionality of the domains.
3. The third step is a proof that f must have a player with no veto power – i.e., that for every value of v_i, the range of f after fixing this value still remains full. This is somewhat in spirit of Barbera and Peleg's proof [2] of Gibbard-Satterthwaite theorem. A critical element is the un-boundedness of the valuation space, shedding some light on the bounded domain example of [11].
4. The fourth step in the proof is the case $n = 2$ (when we already know that there is a player with no veto power). Here we observe that that pricing functions p_i^a mentioned in step 1 satisfy a simple condition that $p_i^a(v_1) - p_i^b(v_1)$ is a (monotone) function only of the scalar $v_1(a) - v_1(b)$. Simple

closed mathematical reasoning implies that real functions that satisfy these conditions must all be linear with the same slope, which directly proves the statement. The critical element in this argument is that the range for every fixed value of v_1 is of size at least 3.

5. The fifth step is induction on n, with the base step being at $n = 2$. The logic is basically that every restriction of a single player results in prices that are linear functions of the remaining players, and since it can be shown that the slopes must be equal for different restrictions, we get the prices must be linear over-all.

2 Preliminaries

We start with some notation. A is the set of alternatives, $|A| \geq 3$. $V = \Re^A$ is the full domain of valuations of a single player. An n-player mechanism is a pair (f, p) where $f : V^n \to A$ and $p : V^n \to \Re$, and $p = (p_1, \cdots, p_n)$, where $p_i : V^n \to R$.

Definition 3. $v^{a+=\delta}$ *the valuation obtained from* v *by increasing the value for* a *by* δ *and not changing any other value.*

In the rest of the paper, unless noted otherwise, i will range over $1, ..., n$; a, b, c, k will always range over A, v_i, v_i' will always range over V, and v_{-i} will range over V^{n-1}.

Definition 4. (f, p) *is* incentive compatible *if for all* i, *all* v_i, v_i' *and all* v_{-i} *we have that* $v_i(f(v_i, v_{-i})) - p(v_i, v_{-i}) \geq v_i'(f(v_i', v_{-i})) - p(v_i', v_{-i})$.

The next two propositions are standard, and hold over all domains not just the full domain:

Definition 5. f *is* weakly monotone *if for all* v_i, v_i' *and* a, b: $f(v_i, v_{-i}) = a$ *and* $f(v_i', v_{-i}) = b$ *implies* $v_i(a) - v_i(b) \geq v_i'(a) - v_i'(b)$.

Lemma 1 (e.g., [12]). *If* (f, p) *is* incentive compatible *then there exist functions* $p_i^a : V^{n-1} \to \Re \cup \{\infty\}$ *such that*

1. *Whenever* $f(v_i, v{-i}) = a$ *we have that* $p_i(v_i, v_{-i}) = p^a(v_{-i})$.
2. $f(v_i, v_{-i}) \in \arg\max_a v_i(a) - p_i^a(v_{-i})$.

Lemma 2 ([7]). *If* (f, p) *is incentive compatible then* f *is weakly monotone.*

Proof. Since f is incentive compatible, we have that $v_i(a) - p_i^a(v^{-i}) \geq v_i(b) - p_i^b(v^{-i})$, since $f(v_i, v_{-i}) = a$. On the other hand, $f(v_i', v_{-i}) = b$, and thus $v_i'(b) - p_i^b(v^{-i}) \geq v_i'(a) - p_i^a(v^{-i})$. Subtracting the two inequalities we get that $v_i(a) - v_i(b) \geq v_i'(a) - v_i'(b)$. □

Definition 6. f *is an* affine maximizer *if there exists constants* $\alpha_i \geq 0$ *and* $\beta_a \in \Re \cup \{\infty\}$ *such that* $f(v_1, ..., v_n) \in \arg\max_a(\sum_i(\alpha_i v_i(a)) - \beta_a)$.

3 Getting Rid of Ties

Definition 7. *f is* strongly monotone *if for all v_i, v'_i and a, b: $f(v_i, v_{-i}) = a$ and $f(v'_i, v_{-i}) = b$ implies $v_i(a) - v_i(b) > v'_i(a) - v'_i(b)$.*

Lemma 3 ([7]). *If for every incentive compatible (f, p) with a strongly monotone f, f is an affine maximizer, then also for every incentive compatible (f, p), f is an affine maximizer.*

4 Existence of No-Veto-Power Players

Definition 8. *Player i is said to hold* no veto power *in f if for every v_i and every a there exists v_{-i} with $f(v_i, v_{-i}) = a$. Player i said to be* decisive *in f if for every v_{-i} and for every a there exists some v_i such that $f(v_i, v_{-i}) = a$.*

Lemma 4. *If (f, p) is incentive compatible and f is strongly monotone then all players, except perhaps a single one, hold no veto power.*

We will prove this by considering the range of v_i (for some fixed player i):

Definition 9. $range(v_i) = \{f(v_i, v_{-i})\}_{v_{-i}}$.

Lemma 5. *If (f, p) is incentive compatible and f is strongly monotone and onto A then $range(\cdot)$ satisfies the following properties:*

1. ***Full Range:*** $\cup_{v_i} range(v_i) = A$.
2. ***Monotonicity:*** $a \in range(v_i)$ *and* $\delta \geq 0$ *implies* $a \in range(v_i^{a+=\delta})$.
3. ***IIA:*** $v_i(a) - v_i(b) = v'_i(a) - v'_i(b)$ *implies that* $range(v_i) \cap \{a, b\} = range(v'_i) \cap \{a, b\}$ *or* $range(v_i) \cap \{a, b\} = \emptyset$.

Proof. We first show that f has a full range. This follows immediately from f being onto A: for each alternative a, let v_i^a, v_{-i}^a be so that $f(v_i^a, v_{-i}^a) = a$. $a \in range(v_i^a)$, and thus $\cup_{a \in A} range(v_i^a) = A$.

For monotonicity, consider v_i such that $a \in range(v_i)$, and v_{-i} be such that $f(v_i, v_{-i}) = a$. By strong monotonicity, for every $\delta > 0$, $f(v_i^{a+=\delta}, v_{-i}) = a$. Hence $a \in range(v_i^{a+=\delta})$.

The proof of IIA is a bit more involved. Let v_i, v'_i be as in the IIA condition. It is enough to show that it cannot be the case the $a, b \in range(v_i)$, $a \in range(v'_i)$, but $b \notin range(v'_i)$. Suppose not. Consider the case where the valuation of each of the other players is identical and defined as follows: $u(b) = 0$, $u(a) = t$, and for each other alternative $k \neq a, b$, $u(k) = -t$, for some t to be defined later. We will show that $f(v, u, \ldots, u) = a$ and $f(v', u, \ldots, u) = b$, and obtain a contradiction to strong monotonicity.

We start by showing that $f(v, u, \ldots, u) = a$. Since $a \in range(v)$, there exist valuations $u'_2, \ldots, u'_n$ such that $f(v, u'_2, \ldots, u'_n) = a$. Choose t to be large enough, so that for every $i \geq 2$, and alternative $k \neq a, b$: $t - u'_i(a) = u(a) - u'_i(a) \geq \max_k u(k) - u'_i(k)$. By strong monotonicity we have that $f(v, u, \ldots, u) = a$.

We now show that $f(v', u, \ldots, u) = b$. Since $b \in range(v')$, there exist valuations $u'_2, \ldots, u'_n$ such that $f(v', u'_2, \ldots, u'_n) = b$. Choose t to be large enough so that: $0 - u'_i(b) = u(b) - u'_i(b) \geq \max u(k) - u'_i(k) = -t - u'_i(k)$, for every alternative $k \neq a$ and $i \geq 2$. We have that $f(v', u, \ldots, u) \in \{a, b\}$. However, $a \notin range(v')$, and thus $f(v', u, \ldots, u) = b$, as needed. □

The rest of the proof considers any $R(\cdot)$ that satisfies these three conditions.

Definition 10. *Alternative a is* dictatable *in $R : V \to 2^A \setminus \{\emptyset\}$ if for some v, $R(v) = \{a\}$.*

Lemma 6. *If $R : V \to 2^A \setminus \{\emptyset\}$ satisfies Full Range (for $|A| \geq 3$), Monotonicity, and IIA then either all alternatives are dictatable in R or none are.*

Proof. The proof consists of the following series of claims.

Claim. For all $v, a, \delta > 0$, either $R(v^{a+=\delta}) = \{a\}$ or $R(v) \subseteq R(v^{a+=\delta}) \cup \{a\}$.

Proof. We show that no alternative is removed from the $R(v^{a+=\delta})$ unless a remains alone in the range. Assume $b \neq a$ remained in $R(v^{a+=\delta})$, and that $c \neq b, a$ was removed. However, this is a contradiction to IIA, since $v(b) - v(c) = v^{a+=\delta}(b) - v^{a+=\delta}(c)$. □

Claim. For all v and for all alternatives a, there exists some $\delta > 0$ such that $a \in R(v^{a+=\delta})$.

Proof. Let v' be such that $a \in R(v')$. Fix some $b \in R(v)$. Assume without loss of generality $v'(a) - v'(b) \geq v(a) - v(b)$ (else, consider $v'^{a+=\gamma}$ instead of v', for sufficiently large $\gamma > 0$, and still have $a \in R(v'^{a+=\gamma})$ by monotonicity). Let $\delta = v'(a) - v'(b) - (v(a) - v(b))$. By Claim 4 either $a \in R(v^{a+=\delta})$ (and we are done), or $b \in R(v^{a+=\delta})$ (since $R(v) \subseteq R(v^{a+=\delta})$). In the latter case, observe that since $v'(a) - v'(b) = v^{a+=\delta}(a) - v^{a+=\delta}(b)$, by IIA and since $b \in R(v^{a+=\delta})$ we also have that $a \in R(v^{a+=\delta})$. □

Claim. Let a be a non dictatable alternative. Let v be such that $a, b \in R(v)$. Let w be such that $v(a) - v(b) \leq w(a) - w(b)$. Then, if $a \in R(w)$ we also have that $b \in R(w)$.

Proof. Let $\delta = w(a) - w(b) - (v(a) - v(b))$. By Claim 4, $a, b \in R(v^{a+=\delta})$ (since a is non-dictatable). The claim now follows by using IIA. □

Claim. Let a be a non-dictatable alternative. For all v, there exists $\delta > 0$ such that $R(v^{a+=\delta}) = A$.

Proof. For alternative k, let w_k be a valuation with $a, k \in range(w)$. Such a valuation exists: Let w'_k be such that $k \in R(w'_k)$. By Claim 4, for some $\delta_k > 0$, $a \in R(w_k^{a+=\delta_k})$. By Claim 4, $k \in R(w_k^{a+=\delta_k})$.

Fix v. For each k, let $r_k = w_k(a) - w_k(k)$. Let $\gamma > 0$ be so that $a \in R(v^{a+=\gamma})$, as guaranteed from Claim 4. Let $\delta \geq \gamma$ be such that $v^{a+=\delta}(a) - v^{a+=\delta}(a) \geq w_k(a) - w_k(k)$, for every k. By Claim 4 $a \in R(v^{a+=\delta})$. By Claim 4 $k \in R(v^{a+=\delta})$, for all k. □

To finish the proof of Lemma 6, suppose there is a dictatable alternative a, and a non-dictatable one b. Let $v, \delta > 0$ be such that $R(v) = \{a\}$, and $R(v^{b+=\delta}) = A$ (as guaranteed by Claim 4). However, for $c \neq b, a$ we have that $v(a) - v(c) = v^{b+=\delta}(a) - v^{b+=\delta}(c)$. By IIA $b \in R(v)$. A contradiction. □

Lemma 7. *If all alternatives of a player are non-dictatable then the player holds no veto power.*

The lemma immediately gives us Lemma 4 since at most one player can have all his alternatives dictatable (otherwise two players will dictate contradicting alternatives).

Proof. Let v be some valuation. Let a, b be some alternatives with $a, b \in R(v)$ (the existence of two such alternatives is guaranteed since all alternatives are non dictatable). By Claim 4, there is some $\delta > 0$ such that $R(v^{a+=\delta}) = A$. For every other $c \neq a, b$, we have that $v(b) - v(c) = v^{a+=\delta}(b) - v^{a+=\delta}(c)$, and thus, by IIA and since $R(v^{a+=\delta}) = A$, we also have that $c \in R(v)$, and hence $R(v) = A$. □

5 Two Players

Lemma 8. *Let $f : V^2 \to A$. If (f, p) is incentive compatible, f satisfies strong monotonicity, and the second player is decisive then f is an affine maximizer.*

Definition 11. *$p : \Re^m \to \Re^m$ is* pair-wise-determined *if $x_a - x_b = x'_a - x'_b$ implies $p^a(x) - p^b(x) = p^a(x') - p^b(x')$. It is* pair-wise-monotone *(decreasing) if $x_a - x_b > x'_a - x'_b$ implies $p^a(x) - p^b(x) \leq p^a(x') - p^b(x')$.*

Claim. If (f, p) is incentive compatible, f satisfies strong monotonicity, and the second player is decisive, then the vector p of payment functions $p^a : \Re^m \to \Re$ associated with it by Lemma 1 is pair-wise-determined and pair-wise-monotone.

Proof. We first note that the function $p^a(x)$ are always finite, as an infinite value of $p^a(x)$ will cause a never to be the value of f contradicting decisiveness. Now assume by way of contradiction to one of these assertions that $p^a(x) - p^b(x) > p^a(x') - p^b(x')$, while $x_a - x_b \geq x'_a - x'_b$. Choosing y such that $p^a(x) - p^b(x) > y_a - y_b > p^a(x') - p^b(x')$, with low values for all other y_c's will give, by Lemma 1, $f(x, y) = b$ but $f(x', y) = a$, contradicting strong monotonicity.

The rest of the proof follows directly from this property:

Claim. Let $p : \Re^m \to \Re^m$ be pair-wise determined and pair-wise-monotone then for some fixed function $h : \Re^m \to \Re$, constant $\alpha \geq 0$ and constants $\beta_a \in \Re$ we have that for all a, $p^a(x) = h(x) - \alpha x_a - \beta_a$.

Claim 5 directly implies the lemma: By Lemma 1, we know that $f(x, y) \in \arg\max_a(y_a - p^a(x))$ and since $h(x)$ does not depend on a, $f(x, y) \in \arg\max_a (y_a + \alpha x_a + \beta_a)$ as required.

Proof. (of Claim 5) For ease of notation, we start by assuming without loss of generality that $p^c(x) = 0$ for all x, where c is some fixed alternative. This is without loss of generality since neither the assumptions nor the result of the lemma changes when subtracting a fixed function p^c from all entries p^a. It now suffices to prove the characterization for x with that $x_c = 0$ since by pair-wise determination adding a constant k to all entries does not change $p^a(x)$, while increasing the right-hand-side by the fixed constant $\alpha \cdot k$ (the same for all a) which can be folded back into $h(x)$.

Definition 12. $\Delta^a(x) = p^a(x^{a+=\delta}) - p^a(x)$.

Claim. For every x, $\Delta^a(x) = \Delta^b(x)$.

Proof. By pair-wise determination applied to a, c we have that $p^a(x^{a+=\delta}) = p^a(x^{a+=\delta,b+=\delta})$ and similarly $p^b(x^{b+=\delta} = p^b(x^{a+=\delta,b+=\delta})$. But then

$$\Delta^a(x) - \Delta^b(x) = (p^a(x^{a+=\delta}) - p^a(x)) - (p^b(x^{b+=\delta}) - p^b(x)) =$$

$$(p^a(x^{a+=\delta,b+=\delta}) - p^b(x^{a+=\delta,b+=\delta}) - (p^a(x) - p^b(x)) = 0$$

where the equality to 0 follows from pair-wise determination applied to a, b. □

Claim. There exists a constant $l = l(\delta)$ such that for all x and a, $\Delta^a(x) = l(\delta)$.

Proof. Using pair-wise determination on a, c, $\Delta^a(x)$ may only depend on $x_a - x_c$, and similarly $\Delta^b(x)$ may only depend on $x_b - x_c$. Since Claim 5 showed that these are equal then for all x, y such that $x_c = y_c$ also $\Delta^a(x) = \Delta^a(y)$. Now take x, y – by pair-wise determination $\Delta^a(x) = \Delta^a(y^{a+=y_c-x_c}) = \Delta^a(y)$, where the last equality is since y and $y^{a+=y_c-x_c}$ have the same c-coordinate. □

We now conclude the proof of Claim 5. By definition we have that $l(\delta + \gamma) = l(\delta) + l(\gamma)$ and so for integer k, $l(k\delta) = k \cdot l(\delta)$, and then also for rational q, $l(q\delta) = q \cdot l(\delta)$. By the definition of pair-wise (decreasing) monotonicity (used here for the only time) applied to a, c we see that $\delta \geq \gamma$ implies $l(\delta) \leq l(\gamma)$. This implies the extension of $l(q\delta) = q \cdot l(\delta)$ to all reals q. Now define $\alpha = -l(1)$ (with $\alpha \geq 0$ since $l(1) < 0$) and we have that for every x, y and a, $p^a(x) - p^a(y) = -\alpha \cdot (x_a - y_a)$. Now define $\beta_a = -p^a(\mathbf{0})$ so $p^a(x) = -\alpha \cdot x_a - \beta_a$ as required. □

6 $n \geq 3$ Players

Theorem 2 ([15]). *Let (f, p) be incentive compatible and onto then f is an affine maximizer.*

Proof. By Lemma 3 we assume with out loss of generality that f is strongly monotone. We now prove the result by induction on n, with our base case $n = 2$ shown in Lemma 8.

Assume correctness for $n - 1$ players. We now prove for n. By Lemma 4, all players, except perhaps player n (without loss of generality) have no veto power, and thus for any fixed value of v_1, the induced function, $f_{v_1}(v_{-1}) = f(v_1, v_{-1})$, is onto so by the induction hypothesis is an affine maximizer $f_{v_1}(v_{-1}) \in \arg\max_a (\sum_i(\alpha_i^{v_1} v_i(a)) - \beta_a^{v_1})$. Without loss of generality we assume that f_{v_1} is normalized: for each v_1 and all i, $\alpha_i \leq 1$, with at least one $\alpha_1^{v_1} = 1$, and that we have $\beta_c^{v_1} = 0$. We now show:

Lemma 9. *The values α_i do not depend on v_1. I.e., for each v_1 and v_1' and i, $\alpha_i^{v_1} = \alpha_i^{v_1'}$.*

Proof. Suppose not. Without loss of generality, v_1 and v_1' differ only in their value for c, and $\alpha_i^{v_1} > \alpha_i^{v_1'}$. Let j be the player with $\alpha_j^{v_1} = 1$. Observe that $\alpha_j^{v_1} \leq \alpha_j^{v_1'}$, since the weights are normalized. Define the following valuations: for player j, $v_j(a) = t$, where $t >> |\beta_a^{v_1} - \beta_b^{v_1}|$, and $v_j(k) = 0$ for all $k \neq a$. For player i, define $v_i(b) = (\alpha_j^{v_1} v_j(b) + (\beta_a^{v_1} - \beta_b^{v_1}) - \epsilon)/\alpha_i$, and $v(k) = 0$ for all $k \neq b$. For each player $l \neq 1, j, k$ let v_l be identically zero. For small enough values of $\epsilon > 0$, we have that $f_{v_1}(v_1, v_2, \ldots, v_n) = a$ but $f_{v_1}(v_1', v_2, \ldots, v_n) = b$, a contradiction to strong monotonicity. □

If there is a player with $\alpha_i = 0$, then the output does not depend on his valuation. In this case f is essentially a function for $n-1$ players, and hence it is an affine maximizer by the induction hypothesis. Else, all $\alpha_i > 0$, and in particular we have that all players, perhaps except the first one, are decisive.

Lemma 10. *For each v_1, v_1', $\beta_a^{v_1} - \beta_a^{v_1'} = \alpha_1(v_1(a) - v_1'(a))$, for every alternative a.*

Proof. We require the following claim first:

Claim. $\beta_a^{v_1}$ depends only on $v_1(a)$.

Proof. Let v_1, v_1' be such that $v_1(a) = v_1'(a)$ and $\beta_a^{v_1} > \beta_a^{v_1'}$. Let $v_2(a) = -(\beta_a^{v_1} - \beta_a^{v_1'})/2$, $v_2(c) = 0$, and for each other $k \neq a, c$, $v(k) = -\max(|\beta_k^{v_1}|, |\beta_k^{v_1'}|) - \epsilon$, for some $\epsilon > 0$. Let all other valuations be identically zero. Now, $f(v_1, v_{-i}) = a$ while $f(v_1', v_{-i}) = c$, a contradiction to strong monotonicity. □

Thus, it is enough to consider identical valuations v_1, v_1' that only differ in their value for a. Consider the following two ways to calculate the output. In the first one, given valuations $v_1, \ldots, v_n$ we calculate the output according to f_{v_1}. In the second one, define $f_{v_2,\ldots,v_{n-1}}(v_1, v_n) = f(v_1, \ldots, v_n)$ and calculate according to $f_{v_2,\ldots,v_{n-1}}$. Since fixing some players and using the same price functions still result in a truthful mechanism, we may assume that in both $f_{v_2,\ldots,v_{n-1}}$ and f_{v_1} the prices are calculated according to the price functions $p_1, \ldots, p_n$ of f. Also notice that by the induction hypothesis both functions are affine maximizers (the range of both is A since player n is decisive). Now, for each alternative a:

$$p_i^a(v_1) - p_i^a(v_1') = \alpha_1(v_1(a) - v_1'(a)) = \Sigma_{i\geq 2}\alpha_i v_i(a) + \beta_a^{v_1} - (\Sigma_{i\geq 2}\alpha_i v_i(a) + \beta_a^{v_1'}) = \beta_a^{v_1} - \beta_a^{v_1'}$$

where the first equality is by calculating the price difference according to $f_{v_2,\ldots,v_{n-1}}$ and using the fact that it is an affine maximizer, the second equality is by calculating the price difference according to f_{v_1} and $f_{v_1'}$ and taking into account that both f_{v_1} and $f_{v_1'}$ are affine maximizers. □

In total we get that the function f_{v_1} maximizes a function of the form $\arg\max_a \Sigma_{i\geq 2}\alpha_i v_i(a) + (\beta_a + \alpha_1 v(a)) = \arg\max_a \Sigma_{i\geq 1}\alpha_i v_i(a) + \beta_a$, hence f is an affine maximizer, as needed. □

Acknowledgements. We thank Itai Ashlagi, Ron Lavi, Dov Monderer, and Ariel Procaccia for helpful discussions.

References

1. Archer, A., Tardos, É.: Truthful mechanisms for one-parameter agents. In: FOCS 2001 (2001)
2. Barbera, S., Peleg, B.: Strategy-proof voting schemes with continuous preferences. Social Choice and Welfare (1988)
3. Bartal, Y., Gonen, R., Nisan, N.: Incentive compatible multi unit combinatorial auctions. In: TARK 2003 (2003)
4. Dhangwatnotai, P., Dobzinski, S., Dughmi, S., Roughgarden, T.: Truthful approximation schemes for single-parameter agents. In: FOCS 2008 (2008)
5. Dobzinski, S., Nisan, N.: Limitations of vcg-based mechanisms. Preliminary version in STOC 2007 (2007)
6. Dobzinski, S., Sundararajan, M.: On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In: EC 2008 (2008)
7. Lavi, R., Mu'alem, A., Nisan, N.: Towards a characterization of truthful combinatorial auctions. In: FOCS 2003 (2003)
8. Lavi, R., Mualem, A., Nisan, N.: Two simplified proofs for roberts theorem. Social Choice and Welfare (2009)
9. Lehmann, D., O'Callaghan, L.I., Shoham, Y.: Truth revelation in approximately efficient combinatorial auctions. JACM 49(5), 577–602 (2002)
10. Mas-Collel, A., Whinston, W., Green, J.: Microeconomic Theory. Oxford university press, Oxford (1995)
11. Moldovanu, B., ter Vehn, M.M.: Ex-post implementation with interdependent valuations. Technical report (2002)
12. Nisan, N.: Introduction to Mechanism Design (for Computer Scientists). In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory (2007)
13. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York (2007)
14. Papadimitriou, C., Schapira, M., Singer, Y.: On the hardness of being truthful. In: FOCS (2008)
15. Roberts, K.: The characterization of implementable choise rules. In: Laffont, J.-J. (ed.) Aggregation and Revelation of Preferences. Papers presented at the first European Summer Workshop of the Economic Society, pp. 321–349. North-Holland, Amsterdam (1979)

Characterizing Incentive Compatibility for Convex Valuations

André Berger, Rudolf Müller, and Seyed Hossein Naeemi

Maastricht University, Department of Quantitative Economics, The Netherlands
{a.berger,r.muller,h.naeemi}@ke.unimaas.nl

Abstract. We study implementability in dominant strategies of social choice functions when sets of types are multi-dimensional and convex, sets of outcomes are arbitrary, valuations for outcomes are convex functions in the type, and utilities over outcomes and payments are quasi-linear. Archer and Kleinberg [1] have proven that in case of valuation functions that are linear in the type monotonicity in combination with a local integrability condition are equivalent with implementability. We show that in the case of convex valuation functions one has to require in addition a property called decomposition monotonicity in order to conclude implementability from monotonicity and the integrability condition. Decomposition monotonicity is automatically satisfied in the linear case.

Saks and Yu [9] have shown that for the same setting as in Archer and Kleinberg [1], but finite set of outcomes, monotonicity alone is sufficient for implementability. Later Archer and Kleinberg [1], Monderer [6] and Vohra [10] have given alternative proofs for the same theorem. Using our characterization, we show that the Saks and Yu theorem generalizes to convex valuations. Again, decomposition monotonicity has to be added as a condition.

Keywords: Mechanism design, Social choice theory, Incentive compatibility, Convexity.

1 Introduction

The main goal of mechanism design is to design mechanisms that motivate the agents with private information to choose equilibrium strategies that lead to an implementation of a desired social choice function. In this paper we assume that players have preferences in terms of monetary valuations and that the mechanism designer can use payments to direct agent behavior. Players are assumed to have quasi-linear utilities over outcomes and payments. Whenever the revelation principle holds, the question of implementability reduces then to the existence of a payment rule such that truth telling becomes an equilibrium, in other words, *lying does not pay*.

In this paper we study conditions on the type spaces of the players and on their valuation functions under which there are easily recognizable properties

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 24–35, 2009.

that characterize truthfully implementable social choice functions, i.e. functions that can be combined with payments that motivate the players to reveal their true type. In particular, the aim is to have payment free, local characterizations, because with such a characterization implementability can be verified without the need to construct payments. In one-dimensional settings such a condition is monotonicity. In multi-dimensional settings (a generalization of) monotonicity is still necessary, but often not sufficient. The goal is then to identify multi-dimensional settings where monotonicity is sufficient, or, if it is not, to find additional necessary conditions, that in combination with monotonicity become sufficient. A well-known condition of this type in case of convex type spaces is path-independence of a particular vector field (see, e.g., Jehiel, Moldovanu and Stacchetti [4] and Müller, Perea and Wolf [7]). Archer and Kleinberg [1] have shown how to replace this condition by a local condition: for every type there exists an open neighborhood such that path-integrals on triangles within this neighborhood are equal to 0. However, their proof requires linear valuation functions. In this paper we show that for the more general case of convex valuation functions the same local integrability condition is sufficient (in combination with monotonicity), if one makes the additional assumption that the allocation rule is decomposition monotone. Müller et al. [7] have shown that in the linear case decomposition monotonicity is satisfied by all monotone allocation rules, which explains why it does not appear explicitly in the theorem of Archer and Kleinberg. Furthermore, Archer and Kleinberg have to make an assumption on the existence of certain integrals. Our proof shows that even in the convex case the additional assumption is satisfied automatically, thus eliminating it also in their setting.

In case of a finite set of outcomes, convex type spaces, and particular linear valuations, Saks and Yu [9] have shown that monotonicity is a sufficient condition for implementability. In other words, path independence is implied by monotonicity. Later Archer and Kleinberg [1], Monderer [6] and Vohra [10] have given alternative proofs for general linear settings[1]. Using our characterization for convex valuations, we show that the Saks and Yu theorem generalizes to convex valuations, again under the additional assumption of decomposition monotonicity. Thereby we provide yet another, but very short proof for the special case of linear valuations. Our proof differs from the proof in Archer and Kleinberg for the linear case in that it uses as an argument for local implementability a convex generalization of a Lemma by Monderer [6].

All our results are stated and proven in terms of single agent models. The characterization result for arbitrary set of outcomes immediately generalizes to a characterization of dominant strategy implementable rules as well as Bayes-Nash

[1] More precisely, Saks and Yu have provided a proof where types are encoded as valuation vectors, with one component for each outcome. Monderer has given a proof for the same encoding, but outcomes are probability vectors over pure outcomes and the range of the social function is finite. Archer and Kleinberg as well as Vohra present a proof for the general setting of linear valuations. The previous proofs can, however, be adopted to the general setting.

implementable rules in the case of multiple agents. The generalization of the Saks and Yu theorem for finite sets of outcomes carries over to dominant strategy implementation in the case of multiple agents.

Organization. Section 2 defines our setting and introduces necessary notation. We prove the main characterization theorem for arbitrary outcome sets and convex valuations in Section 3. In Section 4 we give a short proof of a generalization of the theorem of Saks and Yu [9]. In Section 5 we show how to apply our characterization when outcomes and types are points in $\mathbb{R}^2$ and the valuation for an outcome is the distance to the type.

2 Definitions and Setting

Henceforth we will assume that $T \subseteq \mathbb{R}^d$ $(d \geq 1)$ is a convex set and that $f : T \to A$ is an allocation rule from the set of types T to the set of outcomes A. The valuation for an outcome $a \in A$ of a certain type $t \in T$ is defined by the value $v(a,t)$ given by the function $v : A \times T \to \mathbb{R}$. A *mechanism* is a pair (f,p) of an allocation function f and a payment function $p : T \to \mathbb{R}$. The mechanism is called *truthful* or *incentive compatible* if for all $s,t \in T$ it holds that

$$v(f(s),s) + p(s) \geq v(f(t),s) + p(t), \tag{1}$$

i.e. the utility of a player of type s is always maximized when he reports s. The allocation f is called *truthful* if there exists such a payment function p that makes the mechanism (f,p) truthful. It is our goal to characterize truthful allocation functions without having to provide a p that satisfies (1).

Important concepts in this context are monotonicity and cyclical monotonicity of allocation functions. They can be defined in terms of the absence of negative 2-cycles and negative cycles, respectively, in the type graph T_f, as introduced by Gui et al. [2] and generalized in Archer and Kleinberg [1]. The set of nodes of the type graph is equal to T and every ordered pair of types $s,t \in T$ is connected by a directed edge with edge length either $l_p(s,t)$ or $l_s(s,t)$, which are defined as follows:

$$l_p(s,t) := v(f(s),s) - v(f(t),s), \tag{2}$$

$$l_s(s,t) := v(f(t),t) - v(f(t),s). \tag{3}$$

We call $l_p(s,t)$ and $l_s(s,t)$ the *p-length* and *s-length*, respectively, and use the same terminology for lengths of paths and cycles in the respective graphs.

We can now define monotonicity and cyclical monotonicity for allocation functions.

Definition 1. *An allocation function* $f : T \to A$ *is called* monotone*, if for all* $s,t \in T$ *it holds that* $l_s(s,t) + l_s(t,s) \geq 0$. f *is called* cyclically monotone*, if for all* $k \geq 2$ *and all* $\{s_1,\ldots,s_k\} \subseteq T$*, we have that* $\sum_{i=1}^{k} l_s(s_i,s_{i+1}) \geq 0$*, where indices are taken modulo* k*.*

The p-length and the s-length are related in the sense that the p-length and the s-length of any cycle in T_f is the same.

Property 1. For every $k \geq 2$ and every subset $\{s_1, \ldots, s_k\} \subseteq T$, we have that $\sum_{i=1}^{k} l_p(s_i, s_{i+1}) = \sum_{i=1}^{k} l_s(s_i, s_{i+1})$, where indices are taken modulo k.

Proof. This property follows from the fact that $l_p(s,t) = l_s(s,t) + v(f(s), s) - v(f(t), t)$ for all $s, t \in T$. □

Note that due to Property 1 monotonicity and cyclical monotonicity could have been defined in terms of l_p as well.

It is due to the following result of Rochet that we will concentrate on cyclically monotone allocation functions in the remainder of this paper.

Theorem 1 (Rochet [8]). *An allocation function $f : T \to A$ is truthful if and only if it is cyclically monotone.*

The simple proof employs the fact that, due to our choice of edge lengths, node potentials in the type graph coincide with payment rules that implement the allocation rule. Node potentials exist if and only if the type graph does not have a negative cycle.

In this paper we focus on settings where T is a convex set, and where for all outcomes $a \in A$ the function $v(a, .) : T \to \mathbb{R}$ is convex. That is, for all $s, t \in T$ and $\lambda \in [0, 1]$, $v(a, (1-\lambda)s + \lambda t) \leq (1-\lambda)v(a, s) + \lambda v(a, t)$. Almost all previous literature focused on linear valuation functions. In this case we may identify A with a set of vectors in R^k such that $v(a, t) = a \cdot t$. Saks and Yu [9] choose to present their theorem in the model where $T \subset \mathbb{R}^A$ and $v(a, t) = t_a$, that is, outcomes are unit vectors. Monderer [6] has chosen the same model for T but allowed outcomes to be lotteries over unit vectors. Allocation rules in his model where restricted to those with finite range. Archer and Kleinberg [1], Müller et al. [7] and Vohra [10] allow for arbitrary linear functions. For finite A and linear valuations there are almost no differences between the linear models, for infinite A the canonical representation might move us to infinitely dimensional type spaces. However, even for finite A there is a fundamental difference between linear and convex valuation functions: Moving from a model with convex T and convex valuations to the canonical model may result in a non-convex set in $\mathbb{R}^A$. All previous theorems with linear valuations do not apply on non-convex sets of types. Therefore our results apply to a strictly larger domain of settings. However, one needs an additional condition on the setting which we define next.

Definition 2 (Müller et al. [7]). *Let T be convex. An allocation function $f : T \to A$ is called* decomposition monotone*, if for all $s, t \in T$ and all $\lambda \in [0, 1]$ and we have that:*

$$l_p(s,t) \geq l_p(s, (1-\lambda)s + \lambda t) + l_p((1-\lambda)s + \lambda t, t). \tag{4}$$

It is easy to see that we could have used s-lengths rather than p-lengths in this definition. Müller et al. [7] have shown that for linear valuation functions any monotone allocation rule is decomposition monotone. In the full version of the paper we provide an example that this does not generalize to convex valuation functions.

3 Characterizing Incentive Compatibility

In this section we will prove our main theorem that characterizes cyclically monotone allocation functions for convex valuations and arbitrary outcome sets. We start with two Lemmas that relate s-lengths of edges in the type graph to path integrals on line segments. We will denote by $L_{s,t} := \{s + \lambda(t - s) : \lambda \in [0, 1]\}$ the line segment between two types $s, t \in T$.

Recall that a vector $\nabla \in \mathbb{R}^d$ is a subgradient of a function $h : \mathbb{R}^d \to \mathbb{R}$ at t if $h(s) \geq h(t) + \nabla \cdot (s - t)$ for all $s \in T$. For every $t \in T$ allocation function f defines a convex function $v(f(t), .) : T \to \mathbb{R}$, $s \mapsto v(f(t), s)$. We denote the set of subgradients of $v(f(t), .)$ at $s = t$ by $\partial f(t)$. We assume that $\partial f(t) \neq \varnothing$ on T.[2]

We can now define a vector field $\nabla f : T \to \mathbb{R}^d$ by selecting for each $t \in T$ an element from $\partial f(t)$. Any such vector field satisfies for all $s, t \in T$

$$v(f(t), s) \geq v(f(t), t) + \nabla f(t) \cdot (s - t). \tag{5}$$

We summarize a couple of properties of $\nabla f(t)$ in the following lemma. They are key to the proof of our main theorem.

Lemma 1. *Let $s, t \in T$ and assume that $f : T \to A$ is monotone. Moreover, define $g : [0, 1] \to \mathbb{R}$ by $g(\lambda) = \nabla f(s + \lambda(t - s)) \cdot (t - s)$. Then the following hold:*

1. $\nabla f(s) \cdot (t - s) \leq l_s(s, t) \leq \nabla f(t) \cdot (t - s)$,
2. *g is non-decreasing, and*
3. $\nabla f(s) \cdot (t - s) \leq \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma \leq \nabla f(t) \cdot (t - s)$.

Proof. The first property follows immediately from monotonicity and the definitions of $l_s(s, t)$ and $\nabla f(s)$. For the second property let $0 \leq \lambda_1 < \lambda_2 \leq 1$, and let $r_1 = s + \lambda_1(t - s)$ and $r_2 = s + \lambda_2(t - s)$. Then, by using monotonicity and property 1, we get that

$$\begin{aligned} 0 &\leq l_s(r_1, r_2) + l_s(r_2, r_1) \\ &\leq \nabla f(r_2) \cdot (r_2 - r_1) + \nabla f(r_1) \cdot (r_1 - r_2) \\ &= (\lambda_2 - \lambda_1)(g(\lambda_2) - g(\lambda_1)), \end{aligned}$$

i.e. $g(\lambda_2) \geq g(\lambda_1)$ and the second property is proven.

Since g is non-decreasing, g is integrable on $[0, 1]$ and

$$\begin{aligned} \int_0^1 g(\lambda) d\lambda &= \int_0^1 \nabla f(s + \lambda(t - s)) \cdot (t - s) d\lambda \\ &= \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma. \end{aligned}$$

[2] It is well-known that any convex function on T has a subgradient in all t in the interior of T. We will need the existence also on the boundary of T.

Thus the line integral of ∇f along the path $L_{s,t}$ is well defined and finite[3].

Also, we have that

$$g(0) \leq \int_0^1 g(\lambda)d(\lambda) \leq g(1).$$

If we replace $g(0)$ and $g(1)$ with their respective values, the third property follows. □

In the following we denote for $s_1, s_2, s_3 \in T$, all three distinct, by $\blacktriangle_{s_1,s_2,s_3}$ the convex hull of s_1, s_2, s_3 and let $\triangle_{s_1,s_2,s_3}$ be the path describing the boundary of $\blacktriangle_{s_1,s_2,s_3}$, i.e $L_{s_1,s_2} \cup L_{s_2,s_3} \cup L_{s_3,s_1}$, with direction $s_1 \to s_2 \to s_3 \to s_1$. The following lemma will establish the relation between the line integral of any selection from the subgradient and the s-lengths in the type graph of f.

Lemma 2. *Let $s, t \in T$ and assume that $f : T \to A$ is monotone. For every $n \geq 1$ we let $S_n = \sum_{i=0}^{n-1} l_s(r_i^n, r_{i+1}^n)$, where $r_k^n := s + \frac{k}{n}(t - s)$ for $0 \leq k \leq n$. Then*

$$\lim_{n\to\infty} S_n = \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma.$$

Proof. Fix $n \geq 1$. According to Lemma 1 we have that for $0 \leq i \leq n-1$

$$\nabla f(r_i^n) \cdot (r_{i+1}^n - r_i^n) \leq l_s(r_i^n, r_{i+1}^n) \leq \nabla f(r_{i+1}^n) \cdot (r_{i+1}^n - r_i^n).$$

If we sum up the inequalities we get that

$$\sum_{i=0}^{n-1} \nabla f(r_i^n) \cdot (r_{i+1}^n - r_i^n) \leq S_n \leq \sum_{i=0}^{n-1} \nabla f(r_{i+1}^n) \cdot (r_{i+1}^n - r_i^n).$$

For every $n \in \mathbb{N}$ we define $L_n := \sum_{i=0}^{n-1} \nabla f(r_i^n) \cdot (r_{i+1}^n - r_i^n)$ and $U_n := \sum_{i=0}^{n-1} \nabla f(r_{i+1}^n) \cdot (r_{i+1}^n - r_i^n)$. Since ∇f is line-integrable on the path $L_{s,t}$ we have that

$$\lim_{n\to\infty} L_n = \lim_{n\to\infty} U_n = \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma.$$

Furthermore, since $L_n \leq S_n \leq U_n$, we conclude that

$$\lim_{n\to\infty} S_n = \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma.$$

□

We are now ready to prove our main theorem of this section.

Theorem 2. *Let $T \subseteq \mathbb{R}^d$ be convex. Assume that for every fixed $a \in A$ the function $v(a, .) : T \to \mathbb{R}$ is convex and has non-empty sets of subgradients on T. Assume further that $f : T \to A$ is monotone and decomposition monotone. Then the following are equivalent:*

[3] Archer and Kleinberg [1] make the assumption that the allocation function is locally path integrable in order to get this property. In fact our way of defining ∇f releases us from this assumption.

(1) f is cyclically monotone.
(2) for every $t \in T$ there exists an open neighborhood $U(t) \subseteq \mathbb{R}^d$, $t \in U(t)$, such that for all $s_1, s_2, s_3 \in U(t) \cap T$, all three distinct:

$$\int_{\Delta_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = 0$$

(3) for all $s_1, s_2, s_3 \in T$, all three distinct:

$$\int_{\Delta_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = 0$$

(4) for all $k \geq 3$ and every $\{s_1, \ldots, s_k\} \subseteq T$ and $P = \bigcup_{i=1}^{k} L_{s_i,s_{i+1}}$:

$$\int_{P} \nabla f(\sigma) \cdot d\sigma = 0$$

Proof. **(1) ⇒ (2)** This implication follows immediately from a result in Krishna and Maenner [5]. We provide an elementary proof on the basis of type graphs. Consider any $s_1, s_2, s_3 \in T$, all three distinct. Let ϵ be an arbitrary positive number. From Lemma 2 we get that for $i = 1, 2, 3$ there exist N_i such that for all $n \geq N_i$ we have that

$$S_n^i < \int_{L_{s_i,s_{i+1}}} \nabla f(\sigma) \cdot d\sigma + \frac{1}{3}\epsilon. \tag{6}$$

Now let $N = \max\{N_1, N_2, N_3\}$. For $n \geq N$ it holds that

$$S_n^1 + S_n^2 + S_n^3 < \int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma + \epsilon.$$

Since f is cyclically monotone,

$$S_n^1 + S_n^2 + S_n^3 \geq 0,$$

and thus

$$0 \leq \int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma + \epsilon.$$

Since ϵ is an arbitrary positive number we can conclude:

$$\int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma \geq 0.$$

If we started with $\triangle_{s_1,s_3,s_2}$ we would conclude that

$$\int_{\triangle_{s_1,s_3,s_2}} \nabla f(\sigma) \cdot d\sigma \geq 0.$$

Since $\int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = -\int_{\triangle_{s_1,s_3,s_2}} \nabla f(\sigma) \cdot d\sigma$ we also get that

$$\int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma \leq 0,$$

and thus $\int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = 0$.

(2) ⇒ (3) Let $s_1, s_2, s_3 \in T$. Since $\blacktriangle_{s_1,s_2,s_3}$ is closed and bounded it is compact. According to our assumption, for every point in $\blacktriangle_{s_1,s_2,s_3}$ there is an open neighborhood such that the integral of ∇f along every triangle in the intersection of the neighborhood and T is zero. The Lebesgue Number Lemma implies that there is a $\delta > 0$ such that every subset of $\blacktriangle_{s_1,s_2,s_3}$ of diameter less than δ is contained in at least one of these neighborhoods. In particular, if we subdivide $\blacktriangle_{s_1,s_2,s_3}$ into triangles $\blacktriangle^1, \blacktriangle^2, ..., \blacktriangle^M$ each of which having diameter less than δ, and orient the boarders $\triangle^j$ consistently with $\triangle_{s_1,s_2,s_3}$, we get

$$0 = \sum_{j=1}^{M} \int_{\triangle^j} \nabla f(\sigma) \cdot d\sigma.$$

In this formula, the path-integral of ∇f along $\triangle_{s_1,s_2,s_3}$ appears exactly once. All path-integrals of sides of $\triangle^j$ which are not contained in $\triangle_{s_1,s_2,s_3}$ appear exactly once in each direction of these sides, and cancel each other out. Therefore we have

$$\int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = \sum_{j=1}^{M} \int_{\triangle^j} \nabla f(\sigma) \cdot d\sigma = 0.$$

(3) ⇒ (4) Consider $\{s_1, ..., s_k\} \subseteq T$ and $P = \bigcup_{i=1}^{k} L_{s_i,s_{i+1}}$. P can be decomposed into the following triangles:

$$\triangle_{s_1,s_2,s_3}, \triangle_{s_1,s_3,s_4}, ..., \triangle_{s_1,s_{k-1},s_k}$$

According to our assumption the integral of ∇f along every triangle is zero. By a similar argument as before we get

$$\int_P \nabla f(\sigma) \cdot d\sigma = \int_{\triangle_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma + ... + \int_{\triangle_{s_1,s_{k-1},s_k}} \nabla f(\sigma) \cdot d\sigma = 0.$$

(4) ⇒ (1) Let $k \geq 2$ and $\{s_1, s_2, ..., s_k\} \in T$. Let ϵ be an arbitrary positive number. According to Lemma 2, for every $1 \leq j \leq k$ we have:

$$\exists N_j \text{ such that } \quad \forall n : n \geq N_j \quad S_n^j \geq \int_{L_{s_j,s_{j+1}}} \nabla f(\sigma) \cdot d\sigma - \frac{\epsilon}{k},$$

where $S_n^j = \sum_{i=0}^{n-1} l_s(r_{i,j}^n, r_{i+1,j}^n)$ and $r_{i,j}^n = s_j + \frac{i}{n}(s_{j+1} - s_j)$ for all $0 \leq i \leq n$. Since f is decomposition monotone,

$$S_n^j \leq l_s(s_j, s_{j+1}).$$

So for every $1 \leq j \leq k$

$$\int_{L_{s_j,s_{j+1}}} \nabla f(\sigma) \cdot d\sigma \leq S_n^j + \frac{\epsilon}{k} \leq l_s(s_j, s_{j+1}) + \frac{\epsilon}{k}.$$

If we sum up all these inequalities we get that

$$0 = \int_P \nabla f(\sigma) \cdot d\sigma = \sum_{j=1}^{k} \int_{L_{s_j,s_{j+1}}} \nabla f(\sigma) \cdot d\sigma \leq \sum_{j=1}^{k} l_s(s_j, s_{j+1}) + \epsilon.$$

Therefore

$$\sum_{j=1}^{k} l_s(s_j, s_{j+1}) \geq -\epsilon.$$

Since the last inequality holds for every $\epsilon > 0$, f is cyclically monotone. □

Archer and Kleinberg [1] prove a similar characterization for the case that valuations $v(a, t)$ are linear in t. In particular, we use their approach to show that (2) implies (3). Obviously, in this case $\nabla f(.) = v(f(t), .)$, and it is sufficient to relate path lengths in the type graph to path integrals of $v(f(t), .)$. When applied to their special case of linear valuations, our proof shows that it is not necessary to make the explicit assumption that $v(f(t), .)$ is path integrable.

In Heydenreich et al. [3] it is shown that for any implementable rule f revenue equivalence holds if and only if $dist_p(s, t) = -dist_p(t, s)$ in T_f, where $dist_p(s, t)$ is defined as the infimum over all p-lengths of paths from s to t. By the relation between p-lengths and s-lengths, the same characterization can be stated in terms of distances with respect to s-lengths. From Lemma 2 it follows that

$$dist_s(s, t) \leq \int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma,$$

and therefore $dist_s(s, t) + dist_s(t, s) \leq 0$. By cyclical monotonicity we get

$$dist_s(s, t) + dist_s(t, s) = 0.$$

This proves:

Corollary 1 (Revenue Equivalence). *If T, v and f satisfy the assumptions of Theorem 2 and f is implementable, then any two payments that implement f differ by at most a constant.*

4 A Generalization of Saks and Yu

We will now prove a generalization of the result of Saks and Yu [9] to convex valuation functions.

Theorem 3. *Let $T \subseteq \mathbb{R}^d$ be convex and let $|A|$ be finite. Assume that for every fixed $a \in A$ the function $v(a,.) : T \to \mathbb{R}$ is continuous, convex, and has non-empty sets of subgradients on T. Assume further that $f : T \to A$ is monotone and decomposition monotone. Then f is cyclically monotone.*

This is indeed a generalization of the above mentioned result, since in the case of linear valuation functions every monotone allocation rule is also decomposition monotone [7].

Proof. In order to show that f is cyclically monotone, we will show that condition (2) of Theorem 2 holds for f.

Let $t \in T$. For all $a \in A$ let $D_a := \overline{f^{-1}(a)}$, where $\overline{X}$ denotes the topological closure of a set $X \subseteq \mathbb{R}^d$. Moreover, for all $a \in A$, let $\varepsilon_a(t) := \inf_{x \in D_a} \|x - t\|_2$. Then, for each $a \in A$ we have that $t \in D_a$ if and only if $\varepsilon_a(t) = 0$.

We show first that for each $t \in T$ there exists a neighborhood $U(t)$ of t such that $t \in D_a$ for all $a \in f(U(t))$. Set $A(t) := \{a \in A : \varepsilon_a(t) = 0\}$. As $t \in D_{f(t)}$, we have that $A(t) \neq \emptyset$ and $t \in \bigcap_{a \in A(t)} D_a$. If $A(t) = A$ we let $U(t) = \mathbb{R}^d$, otherwise let $\varepsilon = \min\{\varepsilon_a(t) : a \in A \setminus A(t)\}$. Note that $\varepsilon > 0$. Define $U(t) = \{x \in \mathbb{R}^d : \|x - t\|_2 < \varepsilon\}$.

Next we generalize a lemma by Monderer [6] stating that monotonicity of f on some type set S together with $\bigcap_{a \in f(S)} D_a \neq \varnothing$ implies cyclical monotonicity on S. We prove its generalization to convex valuation for $S = U(t) \cap T$.

For this let $\{s_1, \ldots, s_k\} \subseteq U(t) \cap T$ for some $k \geq 3$. Let us fix $1 \leq i \leq k$. Since $t \in D_{f(s_{i+1})}$, there is a sequence $(t_j)_{j \in \mathbb{N}}$, such that $f(t_j) = f(s_{i+1})$ for every $j \in \mathbb{N}$ and $\lim_{j \to \infty} t_j = t$. Note that

$$\begin{aligned} l_p(s_i, s_{i+1}) &= v(f(s_i), s_i) - v(f(s_{i+1}), s_i) = v(f(s_i), s_i) - v(f(t_j), s_i) \\ &\geq v(f(s_i), t_j) - v(f(t_j), t_j) = v(f(s_i), t_j) - v(f(s_{i+1}), t_j). \end{aligned}$$

Using that $v(a, t)$ is continuous in t we get

$$l_p(s_i, s_{i+1}) \geq v(f(s_i), t) - v(f(s_{i+1}), t).$$

Hence

$$\sum_{i=1}^{k} l_p(s_i, s_{i+1}) \geq \sum_{i=1}^{k} v(f(s_i), t) - v(f(s_{i+1}), t) = 0,$$

and f is cyclically monotone when restricted to $U(t) \cap T$.

Finally, we use Theorem 2 [(1) $\Rightarrow$ (3)] to conclude that for all $s_1, s_2, s_3 \in U(t) \cap T$, all three distinct, we have that $\int_{\Delta_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = 0$. □

5 An Example

In this section we will show an example for an allocation rule to which our result can be applied. Before we give the example, we first give a general class of (non-linear) convex valuation functions that can be used in different contexts.

In this setting we restrict ourselves to the case when the type space as well as the outcome space are a subset of $\mathbb{R}^d$. The valuation functions we consider arise from norms on $\mathbb{R}^d$. This will implicitly mean that agents value those outcomes more that are farther away from their own type. Imagine, for example, that a state council has to decide upon the site for a new garbage dump, and the different communities (agents) want to be as far away as possible from the proposed site.

Lemma 3. *Let $\| \cdot \|$ be a norm on $\mathbb{R}^d$ and let $T \subseteq \mathbb{R}^d$ be convex. Then for any fixed $a \in \mathbb{R}^d$ the valuation function defined by $v(a,t) = \| a - t \|$ is continuous and convex in t.*

We now come to the example in which we use the Euclidean norm for our valuation functions. Suppose $T = A = [0,1]^2$ and define an allocation rule on T by $f(t_1,t_2) = (1-t_1, 1-t_2)$ for every $(t_1,t_2) \in T$. For every t and a in $[0,1]^2$ we define the valuation function $v(a,t)$ as the Euclidean distance of these two points in the plane: $v(a,t) = \|a-t\| = \sqrt{(a_1-t_1)^2+(a_2-t_2)^2}$. In order to have monotonicity, for every s and t in $[0,1]^2$ we must have $\|f(s),s\| + \|f(t),t\| \geq \|f(t),s\|+\|f(s),t\|$. This fact, however, follows easily from the triangle inequality, as the line segments from s to $f(s)$ and from t to $f(t)$ always cross in the "midpoint" $(1/2,1/2)$ of T (cf. Fig. 1 (left)).

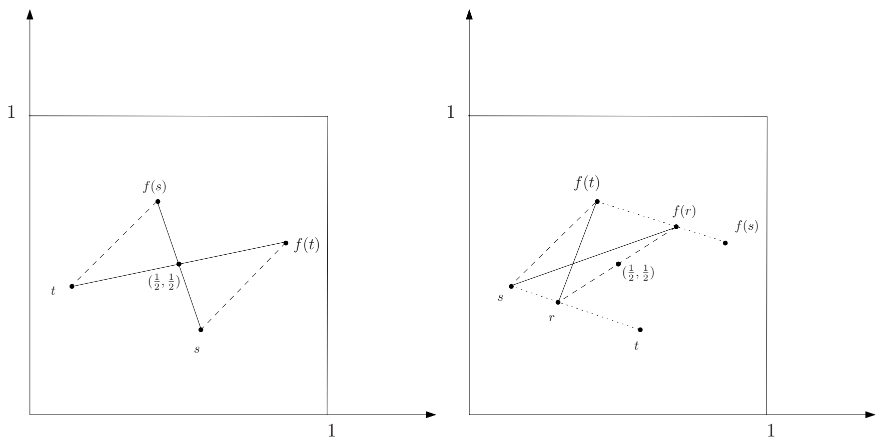


Fig. 1. Left: The segment $(t,f(t))$ passes through $(\frac{1}{2},\frac{1}{2})$ for every $t \in T$. Right: Segments $(s,f(r))$ and $(r,f(t))$ always cross each other.

Decomposition monotonicity of f can be shown similarly (cf. Fig. 1 (right)). Let us now verify the condition from Theorem 2 that will ensure that f is implementable. According to our definition $\nabla f : [0,1]^2 \to [0,1]^2$ is

$$\nabla f(t_1,t_2) = \begin{cases} \left(\frac{-2(1-2t_1)}{\sqrt{(1-2t_1)^2+(1-2t_2)^2}}, \frac{-2(1-2t_2)}{\sqrt{(1-2t_1)^2+(1-2t_2)^2}}\right) & (t_1,t_2) \neq (\frac{1}{2},\frac{1}{2}) \\ (-2,0) & (t_1,t_2) = (\frac{1}{2},\frac{1}{2}). \end{cases}$$

For every s and t in $[0,1]^2$ we get that

$$\int_{L_{s,t}} \nabla f(\sigma) \cdot d\sigma = \sqrt{(1-2t_1)^2 + (1-2t_2)^2} - \sqrt{(1-2s_1)^2 + (1-2s_2)^2}.$$

Since the integral depends only on the end points of $L_{s,t}$ we can conclude that $\int_{\Delta_{s_1,s_2,s_3}} \nabla f(\sigma) \cdot d\sigma = 0$ for all $s_1, s_2, s_3 \in [0,1]^2$. Therefore, according to Theorem 2, f is implementable.

6 Conclusions

In this paper we have presented results about the truthfulness of social choice functions when the valuation functions are assumed to be convex rather than the previously used concept of linear valuations.

Acknowledgements

We thank Benny Moldovanu, Rakesh Vohra, Sascha Wolf and three anonymous referees for valuable comments. The second and the third author acknowledge support by the Hausdorff Center of Mathematics, Bonn.

References

1. Archer, A., Kleinberg, R.: Truthful germs are contagious: a local to global characterization of truthfulness. In: EC 2008: Proceedings of the 9th ACM conference on Electronic commerce, pp. 21–30. ACM, New York (2008)
2. Gui, H., Müller, R., Vohra, R.: Dominant strategy mechanisms with multidimensional types. In: Discussion Paper 1392. The Center for Mathematical Studies in Economics & Management Sciences, Northwestern University, Evanston (2004)
3. Heydenreich, B., Müller, R., Uetz, M., Vohra, R.: Characterization of revenue equivalence. Econometrica 77, 307–316 (2009)
4. Jehiel, P., Moldovanu, B., Stacchetti, E.: Multidimensional mechanism design for auctions with externalities. Journal of Economic Theory 85(2), 258–293 (1999)
5. Krishna, V., Maenner, E.: Convex potentials with an application to mechanism design. Econometrica 69(4), 1113–1119 (2001)
6. Monderer, D.: Monotonicity and implementability: extended abstract. In: EC 2008: Proceedings of the 9th ACM conference on Electronic commerce, pp. 48–48. ACM, New York (2008)
7. Müller, R., Perea, A., Wolf, S.: Weak monotonicity and Bayes-Nash incentive compatibility. Games and Economic Behavior 61(2), 344–358 (2007)
8. Rochet, J.-C.: A necessary and sufficient condition for rationalizability in a quasilinear context. Journal of Mathematical Economics 16(2), 191–200 (1987)
9. Saks, M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proc. 6th ACM conference on Electronic Commerce, pp. 286–293 (2005)
10. Vohra, R.: Paths, cycles and mechanism design. Technical report, Northwestern University (2007)

Truthful Mechanisms for Selfish Routing and Two-Parameter Agents

Clemens Thielen and Sven O. Krumke

Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, D-67663 Kaiserslautern, Germany
{thielen,krumke}@mathematik.uni-kl.de

Abstract. We prove a general monotonicity result about Nash flows in directed networks, which generalizes earlier results and can be used for the design of *truthful mechanisms* in the setting where each edge of the network is controlled by a different selfish agent, who incurs costs proportional to the usage of her edge. Moreover, we consider a mechanism design setting with *two-parameter agents*, which generalizes the well-known setting of one-parameter agents by allowing a fixed cost component as part of each agent's private data. We give a complete characterization of the set of output functions that can be turned into truthful mechanisms for two-parameter agents. This characterization also motivates our choice of linear cost functions without fixed costs for the edges in the selfish routing setting.

Keywords: algorithmic mechanism design, selfish routing, Nash flows.

1 Introduction

The behavior of transportation networks in which the traffic is not controlled by a global authority but by many different selfish users has been studied for decades. As more and more owners of roads (e.g., many European countries) impose tolls on the usage of roads in order to compensate for their costs, the impact of these tolls and the behavior of the owners of the roads on the network traffic becomes a major issue.

Motivated by this setting, we consider a game-theoretic model of transportation networks with two classes of selfish agents. The first class consists of the owners of the network edges and the second class is given by the users of the network. When an edge of the network is used, its owner incurs a cost, which is assumed to be linear in the load on the edge and known only to the owner herself. To compensate for these costs, the owner of an edge imposes a toll on her edge, which every single user of the edge has to pay independently of the load on the edge. Each selfish user tries to choose a path through the network that minimizes the (weighted) sum of the latency she incurs and the overall toll she has to pay to the edges.

To model the interaction of these two classes of selfish users, we use methods from the game-theoretic areas of selfish routing and algorithmic mechanism design. The selfish behavior of the network users is modeled by considering Nash

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 36–47, 2009.

flows. The owners of the network edges are motivated to set the tolls on the edges according to their true per unit costs by using side payments that depend on the load on the edges and the tolls.

In the second part of the paper, we consider an abstract mechanism design setting with *two-parameter agents*. This setting generalizes the well-known case of one-parameter agents by allowing a fixed cost component as part of each agent's private data. In our transportation model, such an additional, privately known cost component could represent a fixed cost for building and maintaining a road, which the owner incurs independently of the load assignment.

Previous Results. Mechanism design is a classical subfield of noncooperative game theory and microeconomics. An introduction to the subject can be found in Chapter 23 of [1]. The systematic study of algorithmic problems in the context of mechanism design was initiated by a seminal paper of Nisan and Ronen [2]. In algorithmic mechanism design, a *mechanism* is defined as a pair $M = (\mathcal{A}, \mathcal{P})$ consisting of an (optimization) algorithm $\mathcal{A}$ and a payment scheme $\mathcal{P}$ defining the side payments to the agents participating in the mechanism. A mechanism is called *truthful with dominant strategies* (in the sequel simply *truthful*) or *strategyproof* if truthtelling is a dominant strategy for every agent, i.e., truthfully revealing her private information maximizes the profit of every agent for any possible behavior of the other agents.

Archer and Tardos [3] considered the important case of algorithmic mechanism design for *one-parameter agents*. In this setting, the type of each agent i is a single nonnegative real number t_i. Each feasible solution x of the global optimization problem results in an amount of *work* $w_i(x)$ being assigned to agent i, who incurs a *cost* of $t_i \cdot w_i(x)$ for completing this amount of work. The profit of agent i is defined as the payment she receives from the mechanism minus her cost. Archer and Tardos [3] showed that an algorithm $\mathcal{A}$ for an optimization problem with one-parameter agents can be used in a truthful mechanism $M = (\mathcal{A}, \mathcal{P})$ if and only if $\mathcal{A}$ is *monotone*, meaning that, for every agent, the amount of work assigned to her does not increase if her bid increases.

Selfish routing is an active research area these days. The book [4] gives a good introduction to the subject. Much work on selfish routing in recent years has focused on quantifying the loss of efficiency due to selfishness. The most common game-theoretic approach is to consider *Nash equilibria*, i.e., solutions of noncooperative games in which no player has an incentive to unilaterally change her strategy. In *nonatomic* models of selfish routing as studied in this paper, the traffic routed by the selfish network users is modeled as a network flow and Nash equilibria are commonly referred to as *Nash flows*. Several authors have recently investigated how Nash flows are influenced by tolls on the network edges [5,6,7].

Our Results. We prove a general monotonicity result about Nash flows in directed networks, which states that the Nash flow on an edge cannot increase when the cost of the edge to the network users is increased. This result generalizes a result of Dafermos and Nagurney [8], who studied an equivalent model of selfish routing. However, the analysis in [8] crucially relies on the so called *strong monotonicity condition* for the cost functions of the network, which is a rather strong assumption and, in particular, requires all cost functions on the network edges to be strictly increasing. We do *not* use the strong monotonicity condition and our monotonicity result holds true in the more general setting of nondecreasing cost functions. Moreover, our result extends to the case of Nash equilibria in *nonatomic congestion games* without modification in the proof.

We use our monotonicity result for the design of truthful mechanisms in the game-theoretic model with two classes of selfish agents described in the introduction. Our result about Nash flows implies that, when considering the toll defined by each owner of an edge as a bid for her privately known cost per unit load, the assignment of load to the edges by a Nash flow yields a *monotone* algorithm, which can be used in a truthful mechanism. Thus, our results connect the research areas of mechanism design and selfish routing, which are two of the main research topics in algorithmic game theory these days.

We motivate the choice of linear cost functions without fixed costs for the edges by proving results about mechanisms for the case of *two-parameter agents* described in the introduction. We show that, for almost all fixed bids for an agent's per unit cost, the load assigned to the agent in a truthful mechanism for two-parameter agents has to be independent of the agent's bid for her fixed cost. Moreover, when the load is continuous in the agent's bid for her per unit cost, it must be completely independent of the agent's bid for her fixed cost, so the situation essentially reduces to the one-parameter setting. Together with the monotonicity of the load assigned to an agent in the agent's bid for her per unit cost as in the one-parameter setting our necessary condition for truthfulness turns out to be sufficient as well, so we obtain a complete characterization of the set of output functions that can be turned into truthful mechanisms for two-parameter agents. Furthermore, we show that no truthful mechanism in the two-parameter setting can satisfy *voluntary participation* (also known as *participation constraints* or *individual rationality constraints*), which means that no truthful mechanism for two-parameter agents can guarantee that agents who bid truthfully never incur a net loss.

2 The Monotonicity of Nash Flows

In the selfish routing part of the paper, we are given a directed network $G = (V, E)$ with vertex set V, edge set E, and K source-destination pairs $(s_1, t_1), \ldots, (s_K, t_K) \in V^2$. $\mathcal{P}_i$ denotes the set of (simple) s_i-t_i paths and is assumed to be nonempty for every $i \in \{1, \ldots, K\}$. We write $\mathcal{P} := \cup_i \mathcal{P}_i$.

A *flow* is a function $F : \mathcal{P} \to \mathbb{R}_{\geq 0}$. For a fixed flow F, we denote the value of F at $p \in \mathcal{P}$ by F_p. Every flow F induces a nonnegative *load* $f_e := \sum_{p \in \mathcal{P}: e \in p} F_p \geq 0$ on every edge $e \in E$, and we call the vector $f = (f_e)_{e \in E}$ the *load vector* of F. Each commodity i has a finite and positive *demand* $d_i > 0$, i.e., d_i units of flow have to be sent from source s_i to destination t_i. Since all demands are finite, we may assume without loss of generality that $\sum_{i=1}^{K} d_i = 1$. A flow F is *feasible* if $\sum_{p \in \mathcal{P}_i} F_p = d_i$ for all $i \in \{1, \ldots, K\}$. A load vector f is *feasible* if it is the load vector of some feasible flow F. The set of all feasible flows will be denoted by D and can be considered as a (compact) subset of $\mathbb{R}^{|\mathcal{P}|}$.

Every edge $e \in E$ is given a nonnegative *cost function* $c_e : [0,1] \ni f_e \mapsto c_e(f_e) \in \mathbb{R}_{\geq 0}$, which specifies the cost for using edge e when the load on e is f_e. We assume the cost functions c_e to be *continuous* and *nondecreasing* and denote the vector of all cost functions by $c = (c_e)_{e \in E}$. The cost of a path $p \in \mathcal{P}_i$ to commodity i is the sum of the costs of the edges in the path, denoted by $c_p(f) = \sum_{e \in p} c_e(f_e)$. We denote the vector of costs of edges $e \in E$ when the load vector is f by $c(f) = (c_e(f_e))_{e \in E}$. We call the triple (G, d, c) an *instance*. In what follows, we will always assume the network G and the demand vector d to be fixed, so an instance is defined by only the vector c of cost functions.

Definition 1. *A feasible flow $F \in D$ with load vector f is* at Nash equilibrium *(or is a* Nash flow*) for costs c if, for every $i \in \{1, \dots, K\}$, the following holds:*

$$c_{p_1}(f) > c_{p_2}(f) \text{ for } p_1, p_2 \in \mathcal{P}_i \text{ implies } F_{p_1} = 0.$$

Flows satisfying the condition of Definition 1 are usually referred to as *Wardrop equilibria* in the literature, but Definition 1 can easily be seen to agree with the usual definition of Nash flows in our setting (cf. [5]). To prove the main theorem of this section (Theorem 1), we need the following results:

Proposition 1. *([9]) A feasible flow F is at Nash equilibrium for costs c if and only if its load vector f satisfies the* variational inequality*:*

$$c(f)^T \cdot (f' - f) \geq 0 \quad \text{for all feasible load vectors } f' \text{ (where "}^T\text{" is transposition)}$$

Proposition 2. *([5]) There exists a Nash flow for every vector c of continuous, nondecreasing cost functions. Moreover, if $F, \tilde{F}$ are Nash flows for costs c and $f, \tilde{f}$ are the respective load vectors, then $c_e(f_e) = c_e(\tilde{f}_e)$ for each edge e. If, in addition, all cost functions c_e are strictly increasing, then $f_e = \tilde{f}_e$ for all $e \in E$.*

Theorem 1. *Let $c, \tilde{c}$ be two vectors of continuous, nondecreasing cost functions, and let $f, \tilde{f}$ be load vectors of Nash flows for costs c and $\tilde{c}$, respectively. If $\tilde{c}_{e_0} \leq c_{e_0}$ for a fixed edge $e_0 \in E$, $\tilde{c}_{e_0}(f_{e_0}) < c_{e_0}(f_{e_0})$, and $\tilde{c}_e = c_e$ for all $e \neq e_0$, then $\tilde{f}_{e_0} \geq f_{e_0}$.*

Proof. By Proposition 1, f and $\tilde{f}$ satisfy the variational inequality, so

$$c(f)^T \cdot (f' - f) \geq 0 \quad \text{and} \quad \tilde{c}(\tilde{f})^T \cdot (f' - \tilde{f}) \geq 0$$

for all feasible load vectors f'. Choosing $f' = \tilde{f}$ in the first inequality and $f' = f$ in the second one and adding yields $(c(f) - \tilde{c}(\tilde{f}))^T \cdot (f - \tilde{f}) \leq 0$.
Setting $\epsilon := c_{e_0}(f_{e_0}) - \tilde{c}_{e_0}(f_{e_0}) > 0$, we obtain:

$$\begin{aligned} 0 &\geq \Big(c(f) - \tilde{c}(\tilde{f})\Big)^T \cdot (f - \tilde{f}) = \textstyle\sum_{e \in E} \Big(c_e(f_e) - \tilde{c}_e(\tilde{f}_e)\Big) \cdot (f_e - \tilde{f}_e) \\ &= \underbrace{\textstyle\sum_{e \neq e_0} \Big(c_e(f_e) - c_e(\tilde{f}_e)\Big) \cdot (f_e - \tilde{f}_e)}_{\geq 0} + \Big(\underbrace{c_{e_0}(f_{e_0})}_{=\epsilon + \tilde{c}_{e_0}(f_{e_0})} - \tilde{c}_{e_0}(\tilde{f}_{e_0})\Big) \cdot (f_{e_0} - \tilde{f}_{e_0}) \\ &\geq \epsilon \cdot (f_{e_0} - \tilde{f}_{e_0}) + \underbrace{\Big(\tilde{c}_{e_0}(f_{e_0}) - \tilde{c}_{e_0}(\tilde{f}_{e_0})\Big) \cdot (f_{e_0} - \tilde{f}_{e_0})}_{\geq 0} \geq \underbrace{\epsilon}_{>0} \cdot (f_{e_0} - \tilde{f}_{e_0}) \end{aligned}$$

Thus, it follows that $\tilde{f}_{e_0} \geq f_{e_0}$ as claimed. □

Note that we only assume the cost functions to be nondecreasing in Theorem 1. The result holds in this setting even though the load vector of a Nash flow is not uniquely defined. Moreover, since Proposition 1 also characterizes Nash equilibria in the more general case of *nonatomic congestion games* (cf. for example [10]), Theorem 1 and its proof extend to this setting.

Also note that, in the case where all cost functions are *strictly* increasing, the assumption $\tilde{c}_{e_0}(f_{e_0}) < c_{e_0}(f_{e_0})$ is not needed: If $\tilde{c}_{e_0}(f_{e_0}) = c_{e_0}(f_{e_0})$, then $\tilde{c}(f) = c(f)$ and it follows from the characterization given in Definition 1 that f is also the load vector of a Nash flow for costs $\tilde{c}$. Hence, the uniqueness implies that $\tilde{f} = f$ and, in particular, $\tilde{f}_{e_0} = f_{e_0}$, so we obtain the following corollary:

Corollary 1. *Let $c, \tilde{c}$ be two vectors of continuous, strictly increasing cost functions, and $f, \tilde{f}$ the unique load vectors of Nash flows for costs c, $\tilde{c}$, respectively. If $\tilde{c}_{e_0} \leq c_{e_0}$ for a fixed edge $e_0 \in E$ and $\tilde{c}_e = c_e$ for all $e \neq e_0$, then $\tilde{f}_{e_0} \geq f_{e_0}$.*

3 Application to Our Model

In our model of transportation networks with two classes of selfish agents, the costs of an edge to the network users are given as a weighted sum of latencies and tolls as follows: For every edge $e \in E$, we are given a nonnegative toll τ_e defined by the owner of edge e and a nondecreasing, continuous latency function $l_e : [0, 1] \to \mathbb{R}_{\geq 0}$. The vector of all tolls is denoted by $\tau = (\tau_e)_{e \in E}$. The cost function c_e of edge $e \in E$ is given by $c_e(x) := l_e(x) + \alpha \cdot \tau_e$, where $\alpha > 0$ is a constant factor describing the sensitivity of the agents to tolls. The total latency on a path $p \in \mathcal{P}_i$ is denoted by $l_p(f) = \sum_{e \in p} l_e(f_e)$, and the total toll on p is denoted by $\tau_p = \sum_{e \in p} \tau_e$. Using this model, Theorem 1 immediately yields:

Corollary 2. *Let the costs be given as a (weighted) sum of latencies and tolls as above, where the latency functions are continuous and nondecreasing. Let $\tau, \tilde{\tau}$ be toll vectors with $\tilde{\tau}_{e_0} < \tau_{e_0}$ for a fixed edge $e_0 \in E$ and $\tilde{\tau}_e = \tau_e$ for all $e \neq e_0$. If $f, \tilde{f}$ are load vectors of Nash flows for tolls τ and $\tilde{\tau}$, respectively, then $\tilde{f}_{e_0} \geq f_{e_0}$.*

Our goal is to design a mechanism that ensures a certain amount of cooperation of both classes of selfish agents, e.g., to make sure that the owners of the edges do not exploit the network users by setting the tolls too high. A *mechanism* in this setting is a pair $(\mathcal{A}, \mathcal{P})$ consisting of an algorithm $\mathcal{A}$, which determines an assignment of load to the edges, and a payment scheme $\mathcal{P}$, which specifies the payments to the edges by the mechanism.

The costs that the owner of an edge e incurs are assumed to be linear in the load on the edge, so they are of the form $t_e \cdot g_e$, where $t_e \geq 0$ is a nonnegative constant, and $g_e \geq 0$ is the load on edge $e \in E$. The per unit cost t_e of the agent controlling edge e is only known to the agent herself. We assume that every agent only controls a single edge in the network and, hence, identify the agent controlling edge $e \in E$ with the edge e itself. The costs of an edge e for the network *users* are different from the costs that the edge incurs due to its usage: Every infinitesimally small, selfish user of commodity i will choose a path p between her source vertex s_i and destination vertex t_i minimizing her cost given by $l_p(g) + \alpha \cdot \tau_p$. Thus, the traffic pattern arising will be a Nash flow with respect to the given latencies and the tolls defined by the edges.

The mechanism considers the toll τ_e defined by edge e as a claimed value (a *bid*) of edge e for t_e. Based on these bids, the mechanism hands out a payment P_e to every edge e in order to motivate the edges to set the tolls according to their true values t_e, which define the costs the edges have to compensate for.

To see how we can use Corollary 2 for the design of a *truthful* mechanism, we first assume the latencies to be *strictly* increasing, so that the load vector of a Nash flow is uniquely determined by the latencies and tolls (cf. Proposition 2). Under this assumption, the situation fits into the framework of mechanism design with one-parameter agents: The agents are the edges, and the private value of edge $e \in E$ is its per unit cost t_e. The selfish behavior of the network users is taken into account by considering Nash flows. Corollary 2 states that the load on an edge of the network cannot increase when the toll on the edge is increased, so the algorithm described above, which just takes the Nash flow with the given latencies and the tolls defined by the edges as the assignment of load to the edges, is a monotone algorithm. Hence, as shown in [3], our mechanism will be truthful if and only if the total amount of money that an edge $e \in E$ receives when the toll vector (bid vector) is τ is given as

$$h_e(\tau_{-e}) + \tau_e \cdot f_e(\tau_{-e}, \tau_e) - \int_0^{\tau_e} f_e(\tau_{-e}, u)du,$$

where the h_e are arbitrary functions, and $f_e(\tau_{-e}, \tau_e) = f_e(\tau)$ denotes the load on edge e in the Nash flow with the given latencies and the tolls τ. τ_{-e} denotes the vector of all tolls except for τ_e.

In our situation, every edge e already gets $\tau_e \cdot f_e(\tau_{-e}, \tau_e)$ units of money from the users traveling on e via the tolls. Thus, the (additional) payment it has to receive from the mechanism in order to obtain a truthful mechanism has to be of the form $P_e(\tau_{-e}, \tau_e) = h_e(\tau_{-e}) - \int_0^{\tau_e} f_e(\tau_{-e}, u)du$. Hence, we obtain:

Theorem 2. *If all latency functions are strictly increasing, the following is a truthful mechanism:*

1. *Let every edge $e \in E$ set the toll τ_e itself.*
2. *Let the selfish users of the network choose their paths themselves, so that a Nash flow is obtained. The users pay the tolls directly to the edges.*
3. *Consider the toll τ_e set by edge $e \in E$ as a bid for the private value t_e of e.*
4. *Hand out the payment $P_e(\tau_{-e}, \tau_e) = h_e(\tau_{-e}) - \int_0^{\tau_e} f_e(\tau_{-e}, u)du$ to edge $e \in E$, where the h_e are arbitrary functions.*

Note that Nash flows in the setting of Theorem 2 can be computed in polynomial time via convex programming (cf. for example [5]).

Also note that truthfulness of a mechanism in our setting implies that the tolls payed by the users of the network are exactly equal to the costs of the edges. Hence, a truthful mechanism ensures that the network users are not exploited by the edges via too high tolls as mentioned at the beginning of this section.

When all latency functions are *linear*, the results of [11] imply that the *total cost* $\sum_{e \in E}(l_e(f_e)f_e + \alpha \cdot \tau_e \cdot f_e)$ (i.e., the average cost experienced by the users of the network) in a Nash flow F with load vector f is at most $\frac{4}{3}$ times that of an optimal flow, i.e., of a feasible flow with minimal total cost. Hence, the total cost of the flow produced by the mechanism from Theorem 2 is at most $\frac{4}{3}$ times optimal in this case. However, a simple example in [11] also shows that without the assumption of linearity of the latencies the total cost of a Nash flow cannot be bounded by any constant factor times the minimal total cost.

We now consider voluntary participation. As proved in [3], a monotone algorithm admits a truthful payment scheme satisfying voluntary participation if and only if, for every e and every fixed vector of bids of all agents except e, the integral of the work curve of agent e is finite, i.e., if $\int_0^\infty f_e(\tau_{-e}, u)du < \infty$ in our setting. It is easy to see that the functions h_e can be chosen such that the mechanism from Theorem 2 has this property under one additional assumption. Namely, we have to assume that, for each commodity i, there exist (at least) two edge disjoint s_i-t_i paths in the network G. Otherwise, there would exist edges that all users of commodity i have to use and the load on each such edge e would be at least d_i no matter what the bid of the edge is. Hence, we would have $\int_0^\infty f_e(\tau_{-e}, u)du \geq \int_0^\infty d_i du = \infty$ in this case. On the other hand, if there are two edge disjoint s_i-t_i paths (say p_1^i and p_2^i) for every commodity i, all users of commodity i will use p_2^i if the toll on an edge $e \in p_1^i$ gets too high, assuming that all other bids (and, thus, all other tolls) are unchanged. Hence, we have $\int_0^\infty f_e(\tau_{-e}, u)du < \infty$ in this case, and the choice $h_e(\tau_{-e}) := \int_0^\infty f_e(\tau_{-e}, u)du$, i.e., $P_e(\tau_{-e}, \tau_e) := \int_{\tau_e}^\infty f_e(\tau_{-e}, u)du$, ensures voluntary participation.

Theorem 3. *The functions h_e can be chosen such that the mechanism described in Theorem 2 satisfies voluntary participation if and only if there exist two edge disjoint s_i-t_i paths in G for every $i = 1, \dots, K$. In this case, we can choose $h_e(\tau_{-e}) := \int_0^\infty f_e(\tau_{-e}, u)du$.*

In the case where the latencies are only assumed to be nondecreasing rather than strictly increasing, there can exist Nash flows for a given toll vector that induce different load vectors. Thus, when we do not assume that the mechanism can make the network users choose their paths according to a certain Nash flow chosen in advance, the mechanism has to deal with the uncertainty about the load assignment resulting from the selfish behavior of the network users. In order to motivate truthful bidding by the edges, the mechanism needs at least *some* information about which load vector will be obtained for a given toll vector since the load vector determines the edges' costs.

In the rest of this section, we show how a *randomized mechanism* truthful in expectation can be obtained in this setting under the assumption that there is a commonly known *probability distribution* of the possible load vectors of Nash flows for every toll vector τ. Here, we define a *randomized mechanism* as a pair $M = (\mathcal{A}, \mathcal{P})$, where $\mathcal{A}$ is a randomized algorithm, which determines a (random) assignment of load to the edges, and $\mathcal{P}$ is a randomized payment scheme, i.e., the payment $\mathcal{P}_e$ to each edge $e \in E$ is a random variable. A randomized mechanism is called *truthful in expectation* if truthtelling maximizes the expected profit of every edge regardless of what the other edges bid and it satisfies *voluntary participation* if the expected profit of an edge bidding truthfully is always nonnegative.

By the results of Archer and Tardos [3], a randomized algorithm $\mathcal{A}$ can be used in a randomized truthful in expectation mechanism if and only if the expected load on each edge is a decreasing function of the edges bid/toll, for every fixed vector of bids/tolls of the other edges. The *expected* payments must then be given by the same formula as in the deterministic case.

Given the probability distribution $\Pr_\tau$ of the possible load vectors of Nash flows for every nonnegative toll vector τ, we can design a randomized mechanism as follows: We let every edge $e \in E$ set the toll τ_e itself and let the network users choose their paths completely by themselves as in the mechanism from Theorem 2. Thus, when the toll vector defined by the edges is τ, every load vector of a Nash flow for tolls τ is obtained with the probability given by $\Pr_\tau$. Hence, we obtain a randomized algorithm for assigning the load to the edges. We now denote the random variable that specifies the load on edge $e \in E$ when the toll vector is τ by $f_e(\tau)$. Corollary 2 then implies that the expected value $E(f_e(\tau))$ of $f_e(\tau)$ is decreasing in the toll/bid of e, so the randomized algorithm can be used in a randomized mechanism that is truthful in expectation. The payments to the edges can be defined by the same formula as in the mechanism from Theorem 2 with f_e replaced by $E(f_e)$. Thus, we obtain the following result:

Theorem 4. *Let all latency functions be continuous and nondecreasing and assume that, for every nonnegative toll vector τ, there is a commonly known probability distribution Pr_τ of the possible load vectors of Nash flows with respect to the tolls τ. Then the following randomized mechanism is truthful in expectation:*
1. *Let every edge $e \in E$ set the toll τ_e itself.*
2. *Let the selfish users of the network choose their paths themselves, so that every load vector of a Nash flow for tolls τ is obtained with the probability given by Pr_τ.The users pay the tolls directly to the edges.*
3. *Consider the toll τ_e set by edge $e \in E$ as a bid for the private value t_e of e.*
4. *Hand out the payment $P_e(\tau_{-e}, \tau_e) = h_e(\tau_{-e}) - \int_0^{\tau_e} E(f_e(\tau_{-e}, u))du$ to edge $e \in E$, where the h_e are arbitrary functions.*

This randomized mechanism ensures cooperation of both classes of selfish agents with the mechanism in the same sense as the mechanism from Theorem 2.

Note that the mechanism does not use randomization to obtain truthfulness or a lower total cost. Randomization is only used to deal with uncertainty about the Nash flow and load assignment resulting from the selfish behavior of the network users in the situation where the mechanism is only given probability distributions over possible load vectors of Nash flows.

When considering voluntary participation, the arguments preceding Theorem 3 immediately yield the following result:

Theorem 5. *The functions h_e defining the payments in the randomized mechanism from Theorem 4 can be chosen such that the mechanism satisfies voluntary participation if and only if there exist two edge disjoint s_i-t_i paths in G for every $i = 1, \ldots, K$. In this case, we can choose $h_e(\tau_{-e}) := \int_0^{\infty} E(f_e(\tau_{-e}, u))du$.*

4 Truthful Mechanisms for Two-Parameter Agents

In this section, we prove our results on truthful mechanisms for two-parameter agents, which also motivate our choice of cost functions in the selfish routing part of this paper.

We consider m agents $1, \ldots, m$. Every agent i has some private data, which is known neither to the mechanism nor to the other agents. Everything except the agents' private data is public knowledge. Agent i's private data is a pair (α_i, β_i) of nonnegative real numbers, also called the agent's *true values*. Each agent reports a *bid* $(a_i, b_i) \in \mathbb{R}^2_{\geq 0}$ for her true values to the mechanism. Based on the vectors a, b given by the bids of the agents, the mechanism's *output algorithm* computes an *output* $o = o(a, b)$, where the *output function* o takes values in a given allowable set O. Note that we do *not* assume the set O of possible outcomes to be finite as is needed for the characterization of truthful mechanisms/social choice functions by the *weak monotonicity condition* [12,13] in more general settings. Each agent i incurs a *cost* $\text{cost}_i(a, b) = \text{cost}_i(o(a, b))$, which depends on her private data and the outcome chosen by the mechanism. To compensate the agents for these costs, the mechanism makes a *payment* $\mathcal{P}_i(a, b)$ to each agent i, which depends on the bids. The objective of every agent i is to maximize her profit given by $\text{profit}_i(a, b) = \mathcal{P}_i(a, b) - \text{cost}_i(a, b)$.

As mentioned earlier, we assume the cost functions of the agents to have a special form: The outcome function o assigns an amount $w_i(a, b) = w_i(o(a, b))$ of *load* or *work* to each agent i and the cost of i is $\text{cost}_i(a, b) = \alpha_i \cdot w_i(a, b) + \beta_i$. That is, the private value α_i measures agent i's cost per unit load and β_i is the fixed cost she incurs independently of the load assigned to her. Our aim is to design *truthful* mechanisms, i.e., mechanisms for which $\text{profit}_i((a_{-i}, \alpha_i), (b_{-i}, \beta_i)) \geq \text{profit}_i((a_{-i}, a_i), (b_{-i}, b_i))$ for all values of (a_{-i}, b_{-i}) and (a_i, b_i). In this setting, a mechanism is a pair $\mathcal{M} = (o, \mathcal{P})$ of an output function o and a vector $\mathcal{P}$ of payment functions. An output function o is said to *admit* a truthful payment scheme if there exist payments $\mathcal{P}$ such that the mechanism $\mathcal{M} = (o, \mathcal{P})$ is truthful.

We now prove our main result on two-parameter agents. The proof extends arguments from the proof of the famous one-parameter monotonicity result in [14].

Theorem 6. *An output function $o = o(a, b)$ admits a truthful payment scheme if and only if, for every i and every pair (a_{-i}, b_{-i}) of vectors of bids of all agents except i, the following holds:*

1. *For every fixed value of b_i, the load $w_i(a, b) = w_i(o(a, b))$ assigned to agent i is nonincreasing in a_i.*
2. *For almost all values of a_i, $w_i(a, b)$ is a* constant *function of b_i (where "for almost all" means "for all but a set of Lebesgue measure zero").*

If these conditions hold, the truthful payments must be given as $P_i(a_i, b_i) = P_i(0, 0) + a_i \cdot w_i(a_i, b_i) - \int_0^{a_i} w_i(x, b_i)dx$, which is independent of b_i for almost all values of a_i.

Proof. "$\Rightarrow$": First assume that there exist payments $\mathcal{P}$ such that the mechanism $\mathcal{M} = (o, \mathcal{P})$ is truthful. We fix an agent i and the other agents' bids (a_{-i}, b_{-i}).

Then, we can consider $\mathcal{P}_i$ and w_i as functions of just agent i's bid (a_i, b_i). We define a function $\mathrm{p}_i : \mathbb{R}^2_{\geq 0} \to \mathbb{R}$ by

$$\mathrm{p}_i(x, y) := \mathcal{P}_i(x, y) - x \cdot w_i(x, y) - y,$$

so $\mathrm{p}_i(\alpha_i, \beta_i) = \mathcal{P}_i(\alpha_i, \beta_i) - \alpha_i \cdot w_i(\alpha_i, \beta_i) - \beta_i$ is the profit of agent i when bidding her true values (α_i, β_i). Truthfulness of the mechanism is then equivalent to

$$\begin{aligned} &\mathrm{p}_i(\alpha_i, \beta_i) \geq \mathcal{P}_i(a_i, b_i) - \alpha_i \cdot w_i(a_i, b_i) - \beta_i \\ \Leftrightarrow\ &\mathrm{p}_i(\alpha_i, \beta_i) \geq \mathcal{P}_i(a_i, b_i) - a_i \cdot w_i(a_i, b_i) - b_i \\ &\qquad\qquad\quad -\alpha_i \cdot w_i(a_i, b_i) + a_i \cdot w_i(a_i, b_i) - \beta_i + b_i \\ \Leftrightarrow\ &\mathrm{p}_i(\alpha_i, \beta_i) \geq \mathrm{p}_i(a_i, b_i) + (\alpha_i - a_i) \cdot (-w_i(a_i, b_i)) + (\beta_i - b_i) \cdot (-1) \end{aligned} \tag{1}$$

for all $\alpha_i, \beta_i, a_i, b_i \geq 0$. In particular, choosing b_i and β_i to be equal in (1) yields

$$\mathrm{p}_i(\alpha_i, b_i) \geq \mathrm{p}_i(a_i, b_i) + (\alpha_i - a_i) \cdot (-w_i(a_i, b_i)) \text{ for all } \alpha_i, a_i, b_i \geq 0.$$

Thus, for fixed b_i, $\mathrm{p}_i(a_i, b_i)$ is a convex function of a_i and $-w_i(a_i, b_i)$ is a subgradient at a_i. Hence, standard results from analysis imply that, for every b_i, $\mathrm{p}_i(a_i, b_i)$ is a continuous function of a_i, differentiable almost everywhere with $\frac{\partial \mathrm{p}_i}{\partial a_i}(a_i, b_i) = -w_i(a_i, b_i)$, and equal to the integral of its derivative. Thus, $\mathrm{p}_i(a_i, b_i) = \mathrm{p}_i(0, b_i) - \int_0^{a_i} w_i(x, b_i) dx$ and by definition of p_i this is equivalent to

$$P_i(a_i, b_i) = P_i(0, b_i) + a_i \cdot w_i(a_i, b_i) - \textstyle\int_0^{a_i} w_i(x, b_i) dx. \tag{2}$$

Since $\mathrm{p}_i(a_i, b_i)$ is convex in a_i for fixed b_i, the partial derivative $\frac{\partial \mathrm{p}_i}{\partial a_i}(a_i, b_i) = -w_i(a_i, b_i)$ is nondecreasing, so $w_i(a_i, b_i)$ is a nonincreasing function of a_i for all $b_i \geq 0$, which proves condition 1. Choosing a_i and α_i to be equal in (1) yields

$$\mathrm{p}_i(a_i, \beta_i) \geq \mathrm{p}_i(a_i, b_i) + (\beta_i - b_i) \cdot (-1) \text{ for all } \beta_i, a_i, b_i \geq 0.$$

Thus, for every fixed a_i, $\mathrm{p}_i(a_i, b_i)$ is a convex function of b_i and -1 is a subgradient at b_i. Again by results from analysis, this implies that, for every a_i, $\mathrm{p}_i(a_i, b_i)$ is continuous in b_i, differentiable almost everywhere, and equal to the integral of its derivative. Moreover, $\frac{\partial \mathrm{p}_i}{\partial b_i}(a_i, b_i) = -1$ whenever p_i is differentiable with respect to b_i, so $\mathrm{p}_i(a_i, b_i) = \mathrm{p}_i(a_i, 0) - b_i$ and

$$\begin{aligned} \mathrm{p}_i(a_i, b_i) &= \mathrm{p}_i(a_i, 0) - b_i = P_i(a_i, 0) - a_i \cdot w_i(a_i, 0) - b_i \\ &= P_i(0, 0) + a_i \cdot w_i(a_i, 0) - \textstyle\int_0^{a_i} w_i(x, 0) dx - a_i \cdot w_i(a_i, 0) - b_i \\ &= P_i(0, 0) - \textstyle\int_0^{a_i} w_i(x, 0) dx - b_i, \end{aligned} \tag{3}$$

so using that $\mathrm{p}_i(a_i, b_i) = P_i(a_i, b_i) - a_i \cdot w_i(a_i, b_i) - b_i$ we obtain

$$P_i(a_i, b_i) - a_i \cdot w_i(a_i, b_i) = P_i(0, 0) - \textstyle\int_0^{a_i} w_i(x, 0) dx.$$

In particular, for $a_i = 0$, we get $P_i(0, b_i) = P_i(0, 0)$ for all $b_i \geq 0$, so by (2) the payments are given by the formula in the claim. Plugging in, we obtain

$$\begin{aligned} \mathrm{p}_i(a_i, b_i) &= \mathrm{p}_i(0, b_i) - \textstyle\int_0^{a_i} w_i(x, b_i) dx = P_i(0, b_i) - b_i - \textstyle\int_0^{a_i} w_i(x, b_i) dx \\ &= P_i(0, 0) - b_i - \textstyle\int_0^{a_i} w_i(x, b_i) dx. \end{aligned}$$

Using (3), this yields

$$\int_0^{a_i} w_i(x,0)dx = \int_0^{a_i} w_i(x,b_i)dx \quad \text{for all } a_i, b_i. \tag{4}$$

Moreover, since $w_i(a_i, b_i)$ is nonincreasing in a_i for every fixed b_i, we have

$$\int_0^{a_i} w_i(x,b_i)dx - \int_0^{\tilde{a}_i} w_i(x,b_i)dx = \int_{\tilde{a}_i}^{a_i} w_i(x,b_i)dx \geq (a_i - \tilde{a}_i) \cdot w_i(a_i,b_i)$$

for all $a_i, \tilde{a}_i \geq 0$, i.e., the function φ defined by $\varphi(a_i) := \int_0^{a_i} w_i(x,b_i)dx$ (which is well-defined by equality (4)) is concave and $w_i(a_i,b_i)$ is a supergradient of φ at a_i for every $b_i \geq 0$. Similar to the convex case, this implies that φ is continuous, differentiable almost everywhere, and equal to the integral of its derivative. Moreover, we have $\varphi'(a_i) = w_i(a_i,b_i)$ for every $b_i \geq 0$ whenever φ is differentiable with respect to a_i. Thus, for almost all a_i, we can differentiate (4) and obtain $w_i(a_i,0) = w_i(a_i,b_i)$ for all $b_i \geq 0$, which proves condition 2.

"$\Leftarrow$": Now suppose that conditions 1. and 2. are satisfied for a given output function o. As before, we fix an agent i and the other agents' bids (a_{-i}, b_{-i}) and consider $\mathcal{P}_i$ and w_i as functions of just agent i's bid (a_i,b_i). We claim that the formula in the claim defines a truthful payment scheme for o. To prove this, we have to show that inequality (1) is satisfied for all $\alpha_i, \beta_i, a_i, b_i \geq 0$, which is equivalent to $p_i(\alpha_i,\beta_i) - \mathcal{P}_i(a_i,b_i) + \alpha_i \cdot w_i(a_i,b_i) + \beta_i \geq 0$ for all $\alpha_i, \beta_i, a_i, b_i \geq 0$. Using that $w_i(\_, b_i)$ is nonincreasing for every b_i by 1. and that $\int_0^{\alpha_i} w_i(x,b_i)dx$ is independent of b_i by 2., we calculate

$$\begin{aligned}
& p_i(\alpha_i,\beta_i) - \mathcal{P}_i(a_i,b_i) + \alpha_i \cdot w_i(a_i,b_i) + \beta_i \\
= \; & \mathcal{P}_i(\alpha_i,\beta_i) - \alpha_i \cdot w_i(\alpha_i,\beta_i) - \beta_i - \mathcal{P}_i(a_i,b_i) + \alpha_i \cdot w_i(a_i,b_i) + \beta_i \\
= \; & \mathcal{P}_i(0,0) + \alpha_i \cdot w_i(\alpha_i,\beta_i) - \int_0^{\alpha_i} w_i(x,\beta_i)dx - \alpha_i \cdot w_i(\alpha_i,\beta_i) - \beta_i \\
& -\mathcal{P}_i(0,0) - a_i \cdot w_i(a_i,b_i) + \int_0^{a_i} w_i(x,b_i)dx + \alpha_i \cdot w_i(a_i,b_i) + \beta_i \\
= \; & (\alpha_i - a_i) \cdot w_i(a_i,b_i) + \int_{\alpha_i}^{a_i} w_i(x,b_i)dx \\
\geq \; & (\alpha_i - a_i) \cdot w_i(a_i,b_i) + (a_i - \alpha_i) \cdot w_i(a_i,b_i) = 0. \qquad \square
\end{aligned}$$

When $w_i(a_i,b_i)$ is continuous in a_i for fixed b_i, we see that the function φ in the proof is differentiable everywhere, so differentiating equation (4) yields the independence of $w_i(a_i,b_i)$ of b_i for every fixed value of a_i. Hence, we obtain:

Corollary 3. *An output function $o(a,b)$ for which the load $w_i(a,b)$ assigned to each agent i is continuous in a_i for every fixed a_{-i}, b admits a truthful payment scheme if and only if $w_i(a,b)$ is independent of b_i and nonincreasing in a_i. In this case, the payments must be given as $P_i(a_i,b_i) = P_i(0,0) + a_i \cdot w_i(a_i,b_i) - \int_0^{a_i} w_i(x,b_i)dx$, which is also independent of b_i.*

As a particular consequence of Theorem 6, no mechanism for two parameter agents can be *strongly truthful*, i.e., make truthtelling the *only* dominant strategy for every agent: Whenever the true per unit cost α_i of an agent i is such that $w_i(\alpha_i,b_i)$ (and, hence, also $\mathcal{P}_i(\alpha_i,b_i)$) is independent of b_i, any bid (α_i,b_i) represents a dominant strategy for agent i.

Furthermore, Theorem 6 implies that the voluntary participation condition can never be satisfied in a truthful mechanism for two-parameter agents: To guarantee a nonnegative profit for every agent i bidding truthfully, we need

$$\text{profit}_i(\alpha_i, \beta_i) = \mathcal{P}_i(\alpha_i, \beta_i) - \alpha_i \cdot w_i(\alpha_i, \beta_i) - \beta_i \geq 0$$

for all $\alpha_i, \beta_i \geq 0$. But by Theorem 6, the value $\mathcal{P}_i(\alpha_i, \beta_i) - \alpha_i \cdot w_i(\alpha_i, \beta_i)$ is independent of β_i for almost every α_i, so the profit of agent i is unbounded from below as $\beta_i \to \infty$ for every such α_i. Thus, without any a priori upper bound for β_i, it is impossible to guarantee a nonnegative profit for agent i when she bids truthfully. Hence, we obtain:

Theorem 7. *The voluntary participation condition can never be satisfied in a truthful mechanism for two-parameter agents.*

References

1. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, Oxford (1995)
2. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior 35, 166–196 (2001)
3. Archer, A., Tardos, É.: Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science, pp. 482–491 (2001)
4. Roughgarden, T. (ed.): Selfish Routing and the Price of Anarchy. MIT Press, Cambridge (2005)
5. Roughgarden, T.: Selfish Routing. PhD thesis, Cornell University (2002)
6. Cole, R., Dodis, Y., Roughgarden, T.: How much can taxes help selfish routing? In: Proceedings of the 4th ACM Conference on Electronic Commerce, pp. 98–107 (2003)
7. Fleischer, L., Jain, K., Mahdian, M.: Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In: Proceedings of the 45th Annual IEEE Symposium on the Foundations of Computer Science, pp. 277–285 (2004)
8. Dafermos, S.C., Nagurney, A.: Sensitivity analysis for the asymmetric network equilibrium problem. Mathematical Programming 28, 174–184 (1984)
9. Smith, M.J.: The existence, uniqueness, and stability of traffic equilibria. Transportation Research 13B, 295–304 (1979)
10. Correa, J.R., Schulz, A.S., Stier-Moses, N.E.: On the inefficiency of equilibria in congestion games. In: Jünger, M., Kaibel, V. (eds.) IPCO 2005. LNCS, vol. 3509, pp. 167–181. Springer, Heidelberg (2005)
11. Roughgarden, T., Tardos, É.: How bad is selfish routing? Journal of the ACM 49(2), 236–259 (2002)
12. Bikhchandani, S., Chatterji, S., Lavi, R., Mu'alem, A., Nisan, N., Sen, A.: Weak monotonicity characterizes deterministic dominant strategy implementation. Econometrica 74(4), 1109–1132 (2006)
13. Saks, M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 286–293 (2005)
14. Archer, A.: Mechanisms for Discrete Optimization with Rational Agents. PhD thesis, Cornell University (2004)

Partition Equilibrium
(Extended Abstract)*

Michal Feldman[1] and Moshe Tennenholtz[2]

[1] School of Business Administration and Center for the Study of Rationality, Hebrew University of Jerusalem, and Microsoft Israel R&D Center
mfeldman@huji.ac.il
[2] Microsoft Israel R&D Center and Technion
moshet@microsoft.com

Abstract. We introduce partition equilibrium and study its existence in resource selection games (RSG). In partition equilibrium the agents are partitioned into coalitions, and only deviations by the prescribed coalitions are considered. This is in difference to the classical concept of strong equilibrium according to which any subset of the agents may deviate. In resource selection games, each agent selects a resource from a set of resources, and its payoff is an increasing (or non-decreasing) function of the number of agents selecting its resource. While it has been shown that strong equilibrium exists in resource selection games, these games do not possess super-strong equilibrium, in which a fruitful deviation benefits at least one deviator without hurting any other deviator, even in the case of two identical resources with increasing cost functions. Similarly, strong equilibrium does not exist for that restricted two identical resources setting when the game is played repeatedly. We prove that for any given partition there exists a super-strong equilibrium for resource selection games of identical resources with increasing cost functions; we also show similar existence results for a variety of other classes of resource selection games. For the case of repeated games we identify partitions that guarantee the existence of strong equilibrium. Together, our work introduces a natural concept, which turns out to lead to positive and applicable results in one of the basic domains studied in the literature.

1 Introduction

When considering a prescribed behavior in a multi-agent system, it makes little sense to assume that an agent will stick to its part of that behavior, if deviating from it can increase its payoff. This leads to much interest in the study of Nash equilibrium in games. A Nash equilibrium is an action profile of the agents for which unilateral deviations are not beneficial. When agents are allowed to use mixed actions, a Nash equilibrium always exists. Moreover, in the context of congestion games [14,12], there always exists a pure action equilibrium. However,

* A full version of this paper including all the proofs is available at the authors' web sites.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 48–59, 2009.

Nash equilibrium does not take into account deviations by non-singleton sets of agents. While stability against deviations by subsets of the agents, captured by the notion of strong equilibrium [3], is a most natural requirement, it is well-known that obtaining such stability is possible only in rare situations. In the context of congestion games, Holzman and Law-Yone [8,9] characterized the networks where strong equilibrium always exist. From pragmatic perspective the most important part of their results is the existence of strong equilibrium in *resource selection games*. In a resource selection game (RSG) we have a set of n players, and a set of m resources. Each player chooses a resource from among the set of resources, and his cost is a non-decreasing function of the number of players who have chosen his selected resource. Needless to say, resource selection games are fundamental and central to work in various communities, such as operations research, computer science, game theory and economics. However, a closer look at the above fundamental result shows severe limitations to its applicability. In particular, the following issues arise:

(1) In the original definition of strong equilibrium a deviation is considered profitable only if it is strictly beneficial to all players. However, it makes much sense to consider super-strong equilibrium, in which a beneficial deviation improves the payoff of at least one of the deviator without hurting any other deviator.
(2) The results on existence of strong equilibrium are obtained for one-shot games, while it makes sense to consider a repeated play, with the desire to have stability against deviations in that game.

As it turns out, the important basic results about resource selection games fail to generalize to either super-strong equilibrium or to repeated resource selection games. Consider the basic setting of two identical resources with (strictly) increasing cost functions. This setting is fundamental to many studies in electronic commerce, operations research, communication networks, and economics. Apparently, there are simple instances of that setting in which there is no super-strong equilibrium, and simple instances of that setting in which there is no strong equilibrium when the game is played repeatedly. In order to deal with these issues, we introduce in this paper the study of *partition equilibrium*, and apply it in the context of resource selection games. Partition equilibrium introduces a social context into the study of group deviations by explicitly stating a partition over the players, allowing only for deviations in which the set of deviators constitutes an element of the partition. Needless to say that partition equilibrium makes much sense in the context of games that take into account the social structure of the set of participants.

One way to view partition equilibrium is as an extension of work on social context games [2]. In a social context game, an agent's utility is effected by the payoffs of its friends, where friends are defined using some topological or graph-theoretic structure. However, unlike previous work on social context games, dealing with single agent deviations, in partition equilibrium we consider the situation where members of a coalition coordinate their activity and potential deviations, as in strong equilibrium. Notice that partition equilibrium suggests a novel solution to *non-cooperative games*; in particular, no side payments are considered or allowed.

Previous work on coalitional congestion games [7,10] has considered side payments in the context of congestion games; in this context each player is a set of agents, each of which is a participant in the resource selection game, and the utility of the player is the sum of his agents' utilities. Side payments however deviate from the non-transferable utility assumption which is the basic assumption in work on strong equilibrium [4,1,5,11,13,6]. Our work on partition equilibrium reconsiders deviations by coalitions in the classical non-transferable utility setting. Notice that in the context of one-shot games, a positive result showing the existence of equilibrium when monetary transfers are allowed implies the existence of super-strong partition equilibrium. Indeed, one of our results can be deduced from these relationships. In most cases however the existence of monetary transfers yields negative results; in fact, even if we have two identical resources with increasing cost functions it has been shown that if coalitions are not restricted to have size of at most two then no equilibrium exists when monetary transfers are allowed; our work shows positive results about the existence of super-strong partition equilibria in this setting, and in much wider sets of resource selection games.

The paper is structured as follows. Section 2 presents some definitions. In particular we define T-SE, strong equilibrium for a partition T, and T-SSE, super-strong equilibrium for a partition T. In section 3 we consider T-SSE for one shot games, and in section 4 we consider T-SE for repeated games. Together, our analysis addresses the above mentioned two basic issues.

In section 3 we first concentrate on resource selection games with increasing cost functions; this is a most classical type of games. Recall that even in the case of two identical resources there is no super-strong equilibrium. We show the existence of T-SSE for any T, and arbitrary number of resources, in that setting. We then extend our results to dealing with the case of two non-identical resources with increasing cost functions, and to the case of two identical resources with non-decreasing (rather than increasing) cost functions. In both cases we provide subtle analysis, yielding positive results about equilibrium existence. Notice that in all related cases these are the first positive results on equilibrium existence when group deviations are considered, and deviations are not required to strictly benefit all agents. We also consider the case of general resource selection games with non-decreasing resources and coalitions bounded by size 2. Since this restricted case is the only one for which a positive result is known in games with monetary transfers (see [10]) we know that a super-strong equilibrium exists in that setting; however, we provide a simpler proof for that case. Unfortunately, as we will illustrate, our results do not scale to arbitrary congestion games.

In section 4 we consider repeated resource selection games. In that setting, strong equilibrium (in the classical sense of Aumann [3]) does not exist even if we have two identical resources with increasing cost functions and we allow deviations of size two. We consider general repeated resource selection games, with non-decreasing cost functions, and show that there exists a T-SE when all elements in the partition are of size at most 2, as well as when all elements in the partition are of size at least 2. The above conditions are in a sense complete: we show the existence of a repeated resource selection game, where the society consists of a singleton and a triplet under which there is no T-SE. In addition, we characterize T-structures that admit a T-SE for a restricted case where the resources are identical and there is a majority of singletons in the partition. In this case we show that if the number of players is odd there is a T-SE if all coalitions are of size at most 3, and that when there is a different coalition structure we can find a resource selection game with no T-SE. If the number of players is even there is a T-SE if all coalitions are of size at most two, and when there is a different coalition structure we can find a resource selection game with no T-SE.

2 Model and Preliminaries

A game is denoted by a tuple $G = \langle N, \{S_i\}_{i=1}^n, \{c_i\}_{i=1}^n \rangle$, where N is the set of players, S_i is a finite action space for player $i \in N$, and $c_i(\cdot)$ is a cost function of player i. We denote by $n = |N|$ the number of players. The action profile space of the players is $S = \times_{i=1}^n S_i$. For an action profile $s \in S$ we denote by s_{-i} the actions of players $j \neq i$, i.e., $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. Similarly, for a set of players Γ (also called a *coalition*) we denote by s_Γ and $s_{-\Gamma}$ the actions of players $j \in \Gamma$ and $j \notin \Gamma$, respectively. The cost function of player i maps an action profile $s \in S$ to a real number, i.e., $c_i : S \to \mathbb{R}$. Throughout this paper we restrict attention to *pure* actions.

Nash Equilibrium (NE): An action profile $s \in S$ is a *pure* Nash Equilibrium if no player $i \in N$ can benefit from unilaterally deviating from his action to another action, i.e., $\forall i \in N \; \forall a \in S_i \; : \; c_i(s_{-i}, a) \geq c_i(s)$.

Resilience to coalitions: A *pure action profile* of a set of players $\Gamma \subseteq N$ specifies an action for each player in the coalition, i.e., $\gamma \in \times_{i \in \Gamma} S_i$. An action profile $s \in S$ is not resilient to a pure *strong* deviation of a coalition Γ if there is a pure action profile γ of Γ such that $c_i(s_{-\Gamma}, \gamma) < c_i(s)$ for every $i \in \Gamma$ (i.e., the players in the coalition can deviate in such a way that *each* player reduces its cost). In this case we say that the coalition Γ has a strongly-profitable deviation.

Definition 1. *A* strong equilibrium (SE) *is a profile that is resilient to a pure strongly-profitable deviation of any coalition $\Gamma \subseteq N$.*

An action profile $s \in S$ is not resilient to a pure *weak* deviation of a coalition Γ if there is a pure action profile γ of Γ such that $c_i(s_{-\Gamma}, \gamma) \leq c_i(s)$ for every $i \in \Gamma$, and $\exists i \in \Gamma$ s.t. $c_i(s_{-\Gamma}, \gamma) < c_i(s)$ (i.e., the players in the coalition can deviate in such a way that none of the players increases its cost, and at least one player strictly reduces its cost). In this case we say that the coalition Γ has a weakly-profitable deviation.

Definition 2. *A* super strong equilibrium (SSE) *is a profile that is resilient to a pure weakly-profitable deviation of any coalition $\Gamma \subseteq N$.*

Note that the set of super strong equilibria is contained in the set of strong equilibria.

Suppose the coalitional structure is exogenously given. That is, the finite set of coalitions is given by a partition $T = (T_1, \ldots, T_k)$ of the set of players. Given a partition T, we shall define the following.

Definition 3. *A* T-strong equilibrium (T-SE) *is a profile that is resilient to a pure strongly-profitable deviation of any coalition* $T_i \in T$*.*

Definition 4. *A* T-super strong equilibrium (T-SSE) *is a profile that is resilient to a pure weakly-profitable deviation of any coalition* $T_i \in T$*.*

Observation 1. *Every SE is also a T-SE for any T, and every SSE is a T-SSE for any T.*

It is important to note that while the set of SE is contained in the set of NE, the set of T-SE (or T-SSE) is not necessarily contained in the set of NE (nor does the set of NE contained in the set of T-SE (or T-SSE)). It might be the case that a single player can deviate unilaterally and strictly improve his own payoff, but if such a deviation reduces the payoff of a member of his coalition (or does not improve it), it will not be considered as a beneficial deviation.

We identify two extreme cases:

Single Coalition Case: where there is a single coalition that contains all of the players; i.e., $T = \{N\}$. In the single-coalition case the set of T-SSE outcomes coincides with the set of Pareto-optimal outcomes; thus there always exists a T-SSE.

Claim. Every finite game admits a T-SSE if $T = \{N\}$.

Fully Distributed Case: This is the case in which each individual player constitutes a coalition; i.e., $T = \{\{1\}, \ldots, \{n\}\}$. In the fully-distributed case the set of T-SE coincides with the set of T-SSE and with the set of NE. Thus, any game that admits a pure NE admits a T-SSE as well.

A direct corollary of the above observations is that every 2-player game that admits a pure NE admits a T-SSE for any T. If it is the single-coalition case, a T-SSE exists by Claim 2, and if it is the fully-distributed case, a T-SSE exists since a NE always exists. This is interesting, for example, in the context of potential (or congestion) games, where many of the counter examples refuting the existence of a SE are 2-player games (see, e.g., SE in cost-sharing connection games [4]). Yet, these games always admit some T-SSE.

2.1 Resource Selection Games

A resource selection setting is characterized by the tuple $\langle M, N, \{b_i(\cdot)\}_{i=1}^{m}\rangle$, where $M = \{M_1, \ldots, M_m\}$ is the set of resources, $N = \{1, \ldots, n\}$ is the set of players (jobs) and $b_i(l) \in \mathbb{R}$ is the cost of resource M_i under a load of l players. We also denote the cost function of resource M_i by a vector

$b_i = (b_i(1), b_i(2), \ldots, b_i(n))$. A resource selection setting has identical resources if $\forall i, i' \in \{1, \ldots, m\}$ $\forall l \in \{1, \ldots, n\}$ $b_i(l) = b_{i'}(l)$. In identical resources settings we will use the vector $b = (b(1), \ldots, b(n))$ to denote the cost vector of all the resources.

A *one-shot resource selection game* (RSG) has N as the set of players, and we identify the set of resources with the set of actions; i.e., the action space S_J of player $J \in N$ are all the individual resources, i.e., $S_J = M$ $\forall J \in N$. The action profile space is $S = \times_{J=1}^{n} S_J$. In an action profile $s \in S$ player J selects resource s_J as its action. The load of a resource M_i in the action profile $s \in S$, denoted $l_i(s)$, is the number of players that chose resource M_i. The cost of a player J who chose resource M_i under profile s is $c_J(s) = b_i(l_i(s))$.

We assume that the cost function $b_i(\cdot)$ of all resources is non decreasing; thus, if $b_i(l) < b_i(l')$ then $l < l'$. In some cases, we will assume a strictly-increasing cost function; i.e., $\forall i \forall l$ $b_i(l) < b_i(l+1)$. In this case, $b_i(l) \leq b_i(l')$ implies $l \leq l'$

Every RSG is a congestion game, thus admits a NE in pure actions. In addition, it has been shown in [8] that every RSG with non-decreasing cost functions admits a SE. Therefore, by Observation 1 it also admits a T-SE for any T. Yet, as we shall see, an RSG with non-decreasing cost functions might not admit a T-SSE, nor shall a repeated RSG necessarily admit a T-SE. These two matters shall be our focus in the following two sections, respectively.

3 T-Super Strong Equilibrium (T-SSE) Existence

Every RSG admits a SE [8], and by Observation 1 admits a T-SE as well. However, an RSG might not admit any SSE. This non-existence may occur even for an RSG with two identical strictly-increasing resources, as the following observation shows.

Observation 2. *There exists a one-shot RSG with two identical strictly increasing resources that does not admit any SSE.*

Proof. Consider an RSG with two identical resources of cost function $b = (1, 2, 3)$ and 3 players $N = \{1, 2, 3\}$. If all three players share the same resource, this is obviously not a SSE (and not even a NE). Suppose WLOG that players 1, 2 are assigned to M_1 and player 3 is assigned to M_2. Then, players 1 and 2 can deviate such that player 1 migrates to M_2, incurring the same cost as before, while player 2 reduces its cost from 2 to 1. Therefore, this game does not admit a SSE. □

3.1 The Case of Identical, Strictly-Increasing Resources

While a SSE might not exist even under the restricted setting of two identical strictly increasing resources, as we shall soon show, every RSG with identical strictly-increasing resources admits a T-SSE for any T. Before formulating the theorem, we introduce the following lemma and definition we will use in the proof of Theorem 3.

Lemma 1. *Let G be an RSG with m identical strictly increasing resources, and let s be a NE of G. Suppose there is a coalition Γ that can weakly improve by deviating to a profile $s' = (s'_\Gamma, s_{-\Gamma})$. It holds that $l_i(s') \leq l_i(s) + 1 \ \forall i \in \{1, \ldots, m\}$.*

Definition 5. *Let $l(s) = (l_1(s), \ldots, l_m(s))$ be the congestion vector of a profile s, sorted in non-increasing order. A* T-spread-out-s *assignment is an assignment obtained by filling out the resources by spreading out the members of each coalition by non-increasing order of $|T_i|$ on the resources, according to the sorted vector $l(s)$.*

Theorem 3. *Every RSG with identical strictly increasing resources admits a T-SSE for any T.*

Proof. Let s be a NE of G. We claim that a T-spread-out-s assignment is a T-SSE. Suppose by way of contradiction there is a weakly profitable deviation of some coalition to a profile s'. Since the resources are identical and strictly increasing there must exist L s.t. $l_i(s) \in \{L, L+1\} \ \forall i$. Denote by k the number of resources of load L. Players assigned to resources of load L and $L+1$ are denoted "low" and "high" players, respectively.

We first claim that $l_i(s') \geq L \ \forall i$. Suppose by way of contradiction that $\exists i$ s.t. $l_i(s') \leq L - 1$. Then, in order to assign all the low jobs, there must exist k additional resources of load at most L. But then it must hold that $\exists i$ s.t. $l_i(s') \geq L + 2$, contradicting Lemma 1. We conclude that $l_i(s') \in \{L, L+1\} \ \forall i$. Since the total number of players remains the same, there must exist k resources of load L and $m - k$ resources of load $L + 1$ in s'.

For every low job J, it must hold that $l_i(s') \leq L$, thus, for every high job it must hold that $l_i(s') = L+1$. Therefore, no job in the coalition strictly improves its load, and the statement follows. □

3.2 The Case of Two Strictly-Increasing Non-identical Resources

In the following few paragraphs we consider the case of two resources, but move to the more general case of non-identical resources. We begin with several characteristics of RSG's with two resources.

We distinguish between two types of deviations by a coalition Γ on two resources, namely *uni-directional* and *bi-directional* deviations. In a uni-directional deviation, some jobs in Γ deviate from one resource to the second one. In a bidirectional deviation, a set Γ_1 ($|\Gamma_1| > 0$) deviates from M_1 to M_2 and a set Γ_2 ($|\Gamma_2| > 0$) deviates from M_2 to M_1, s.t. $\Gamma = \Gamma_1 \cup \Gamma_2$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$. Lemmata 2 through 4 below are used in the proof of Theorem 4.

Lemma 2. *Let G be an RSG with two strictly increasing resources with cost functions $b_1(\cdot)$ and $b_2(\cdot)$, and let s be a NE of G. Suppose WLOG that $b_1(l_1(s)) \leq b_2(l_2(s))$. Suppose there is a* bidirectional *coalition Γ that has a weakly-profitable deviation to a profile $s' = (s'_\Gamma, s_{-\Gamma})$. Then, the coalition must be of the following structure: it should include the set $S = \{J | s_J = M_1, J \in \Gamma\}$ (which deviate to M_2) and $|S| + 1$ coalition members from M_2.*

Lemma 3. *Let G be an RSG with two strictly increasing resources, and let s be a NE of G. Suppose there is a unidirectional deviation by coalition $\Gamma \subset T_i$ that has a weakly-profitable deviation to a profile $s = (s'_{\Gamma}, s_{-\Gamma})$. Then, $s_J = s_{J'}\ \forall J, J' \in T_i$.*

Definition 6. *A vector $(\sigma_1, \sigma_2, \dots \sigma_m)$ is* lexicographically smaller than $(\hat{\sigma}_1, \hat{\sigma}_2, \dots \hat{\sigma}_m)$ if for some i, $\sigma_i < \hat{\sigma}_i$ and $\sigma_k = \hat{\sigma}_k$ for all $k < i$.

An action profile s is cost-wise lexicographically smaller than s' if the cost vector $c(s) = (c_1(l_1(s)), \dots, c_m(l_m(s)))$, sorted in non-increasing order, is smaller lexicographically than $c(s')$, sorted in non-increasing order. We denote this relationship by $s \prec s'$.

Lemma 4. *Let s be a cost-wise lexicographic minimum of an RSG game G with non-decreasing resources. Then, s is a NE of G.*

Theorem 4. *Any RSG with two strictly increasing resources admits a T-SSE for any T.*

3.3 The Case of m Non-decreasing Non-identical Resources

We next consider the more general case of non-decreasing cost functions. We show that if $|T_i| \leq 2\ \forall i$ a T-SSE always exists. While this theorem follows as a special case of [10], we choose to present it due to the simplicity of the proof. Before formulating our theorem, we present the following lemma.

Lemma 5. *Let G be an RSG with m non-decreasing resources, and let s be a NE of G. Given a coalition T_j, if it holds that $|\{J | J \in T_j, s_J = M_i\}| \leq 1\ \forall i$, then T_j has no weakly profitable deviation.*

With this we are ready to state the theorem.

Theorem 5. *Every RSG G with non-decreasing resources in which $|T_i| \leq 2\ \forall i$ admits a T-SSE.*

Proof. Let s be a NE of G and consider a T-spread-out-s assignment (we abuse notation and use s to denote both the Ne and the T-spread-out NE). In s, there might be only a single resource that contains more than a single member of each coalition, denote it M_i. Since s is a NE, no singleton can deviate. By Lemma 5, no coalition (of size 2) that is assigned to different resources can deviate either. Thus, we should only consider deviations of pairs (recall $|T_i| \leq 2\ \forall i$) that are assigned to the same resource. Suppose jobs J, J' are assigned to M_i and let s' be a weakly profitable deviation. If both players deviate, it contradicts s being a NE. Thus, we should only consider a deviation in which $s'_{J'} = M_i$ and $s'_J = M_k$, $k \neq i$.

It must hold that $c_J(s') \leq c_J(s)$, thus $b_k(l_k(s')) = b_k(l_k(s)+1) \leq b_i(l_i(s))$, and by s being a NE it must hold that $b_k(l_k(s)+1) \geq b_i(l_i(s))$; thus $b_k(l_k(s)+1) = b_i(l_i(s))$. Therefore, the cost of player J' must strictly reduce; i.e., $b_i(l_i(s)-1) < b_i(l_i(s))$.

We next claim that s' is also a NE. To show this we show that s' is also a NE. By $b_k(l_k(s)+1) = b_i(l_i(s))$ a unilateral deviation from M_i to M_k or in the other direction are not profitable. A unilateral deviation from $M_l, l \neq i, k$ to M_i is not profitable either since by s being a NE, a job from M_l cannot improve by deviating to M_k, thus by $b_k(l_k(s)+1) = b_i(l_i(s))$ it cannot improve by deviating to M_i either. By s being a NE, it follows that all other unilateral deviations are not profitable either. A similar argument shows that after each deviation of a pair that is assigned to M_i, we should again consider only such deviations. But this process is limited by the number of pairs that are assigned to M_i, which is finite. We conclude that this process must converge to a T-SSE. □

3.4 The Case of Two Non-decreasing Resources

We now turn to another extension. In this subsection we consider the existence of T-SSE for two identical resources, but allow the resources to have general non-decreasing cost functions.

The following lemma provides a condition that must be satisfied in order for a unilateral deviation to occur. It holds for non-identical resources too, and has been used in section 3.2 as well.

Lemma 6. *Let G be an RSG with two non-decreasing resources with cost functions $b_1(\cdot)$ and $b_2(\cdot)$, and let s be a cost-wise lexicographic minimum, T-spread-out NE of G with loads $l_1(s)$ and $l_2(s)$ respectively. A unidirectional deviation of $\Gamma \subseteq T_i$ is possible only from a resource that contains all the members of T_i.*

Lemma 7. *Let G be an RSG with two non-decreasing identical resources, and let s be a NE of G. If there exists a weakly profitable bi-directional deviation, there also exists a weakly profitable uni-directional deviation of a smaller coalition.*

Lemmata 6 and 7 are used to prove the following theorem.

Theorem 6. *Every RSG with two non-decreasing identical resources admits a T-SSE for any T.*

4 T-Strong Equilibrium (T-SE) Existence in Repeated RSG

Consider a *one-shot game* G and an integer R. We define the repeated game $\hat{G} =< G, R >$ as R plays of G, where in period t players choose strategies $(s_1, \ldots, s_n) \in (S_1, \ldots, S_n)$ after observing the actions taken by all the users in all previous periods. A strategy of player J is a function p_J specifying the action of player J at time t, given the history up to time $t-1$. Player J's cost in $\hat{G}$ is $c_J(p) = \sum_{t=1}^{R} c_J(s_1(t), \ldots, s_n(t))$, where $s_J(t)$ is player J's one-shot game action in period t according to p_J.

There is a crucial difference between SE and NE in repeated games. Suppose s is a NE of the game G. Then, playing s in every round of the repeated game must be a NE of the repeated game. In contrast, if s is a SE of the game G, it is not necessarily the case that playing s in every round of the repeated game is a SE of the repeated game.

For example, while every RSG admits a SE, even on the very simple RSG that is composed of two identical resources with non-decreasing cost functions and 3 players, its repeated version might not admit a SE.

Observation 7. *[15]. There exists a repeated RSG with two identical non-decreasing resources and 3 players that does not admit a SE.*

Similarly, if s is a T-SE of the game G, it is not necessarily the case that playing s in every round of the repeated game is a T-SE of the repeated game (as exemplified by Theorem 10). Thus, characterizing the set of repeated games that admit a T-SE is a challenging goal.

4.1 The General Case

The following theorem shows that every repeated RSG on m non-decreasing resources admits a T-SE if T contains no singletons. We first define a Γ-minimal player and present several lemmas that will be used in the proof of the theorem.

Definition 7. *Let G be a one-shot game and let s be an action profile in G. Player i is said to be Γ^s-minimal if for any action profile $s' = (s'_\Gamma, s_{-\Gamma})$ it holds that $c_i(s) \leq c_i(s')$.*

Lemma 8. *Let $\hat{G} = \langle G, R \rangle$, and let s be a strategy profile s.t. $\forall T_i \ \exists J \in T_i$ s.t. J is T_i^s-minimal $\forall r \in R$. Then, playing s in every round of $\hat{G}$ is a T-SE of $\hat{G}$.*

The following theorems identify a family of T-structures for which a T-SE always exists.

Theorem 8. *Every repeated RSG with m non-decreasing resources admits a T-SE if $|T_i| \geq 2 \ \forall i$.*

In addition, every repeated RSG on m non-decreasing resources admits a T-SE if $|T_i| \leq 2 \ \forall i$. We first introduce a lemma that will be used here and in what follows.

Lemma 9. *Let G be an RSG with m non-decreasing resources, and let s be a NE of G s.t. $\forall i \ \forall j \ |\{J | J \in T_j, s_J = M_i\}| \leq 1$ (i.e., no resource contains more than a single representative of each coalition). Then, playing s in every round constitutes a T-SE of the repeated game.*

With this we are ready to establish the theorem.

Theorem 9. *Every repeated RSG with m non-decreasing resources admits a T-SE if $|T_i| \leq 2 \ \forall i$.*

In addition, the above conditions are tight. In particular, there exists a repeated RSG that does not adhere to the structure described above, which does not admit a T-SE, as the following theorem shows.

Theorem 10. *There exists a repeated RSG with two identical non-decreasing resources s.t.* $|T_1| = 1$ *and* $|T_2| = 3$ *that does not admit a* T*-SE.*

Proof. Let G be an RSG with two identical resources with cost function $b(\cdot)$ and four players, where $T = \{T_1, T_2\}$, s.t. $|T_1| = 3$ and $|T_2| = 1$, $b(1)+2b(3) < 3b(2)$, and $b(2) < b(3)$. Consider the game $\hat{G} = \langle G, 3\rangle$. Suppose by way of contradiction that the repeated game above admits a T-SE s. In the third (and last) stage of the game, the singleton can never share a resource with more than one additional player, since if it does, it incurs a cost of at least $b(3)$ and by deviating it can incur a cost of at most $b(2)$. Second, the three players in T_1 cannot all share a resource, since if one of them deviates, all three players reduce their cost from $b(3)$ to $b(2)$. Therefore, in the third stage, every resource should be assigned exactly two players. Using a backward induction argument, under the profile s, in every stage of the game two players should be assigned to every resource, i.e., $l_i(s) = 2\ \forall i \in M$. Consider the following deviation s' of T_1: each player in T_1 is left alone in one of the stages and has a load of 3 in the other two stages. For every player $J \in T_1$, it holds that $c_J(s') = 2c(3) + c(1) < 3c(2) = c_J(s)$. Therefore, s' is a strongly-profitable deviation of T_1 and the game admits no T-SE. □

The construction given in the proof of Theorem 10 implies that the existence of T-SE in one-shot RSGs does not apply to general congestion games. This can easily be verified by constructing a congestion game that consists of three networks that are composed serially, where each network is composed of two parallel edges with the cost functions and T-structure given in the example above.

4.2 The Case of Majority of Singletons

For special cases, we have a more refined characterization. In particular, if the majority of the players are singletons and the resources are identical, we fully characterize the T-structures that serve as the condition for the existence of T-SE in repeated games. The characterization is slightly different for odd and even number of players.

Odd Number of Players. For an odd number of players, we find that $|T_i| \leq 3$ is a sufficient and necessary condition for the existence of T-SE, as shown in the following theorems.

Theorem 11. *For every* T *s.t.* $\exists i$ *s.t.* $|T_i| \geq 4$*, there exists a repeated RSG on identical resources with an odd number of players and a majority of singletons that does not admit a* T*-SE.*

Theorem 12. *For every repeated RSG* $\hat{G}$ *on* m *identical non-decreasing resources with a majority of singletons and an odd number of players, if* $|T_i| \leq 3\ \forall i$*,* $\hat{G}$ *admits a* T*-SE.*

Even Number of Players. For an even number of players, we find that $|T_i| \leq 2$ is a sufficient and necessary condition for the existence of T-SE, as shown in the following theorems.

Theorem 13. *For every T s.t. $\exists i$ s.t. $|T_i| \geq 3$, there exists a repeated RSG on identical resources with an even umber of players and a majority of singletons that does not admit a T-SE.*

Theorem 14. *For every repeated RSG on m identical non-decreasing resources with a majority of singletons and an even number of players, if $|T_i| \leq 2$ $\forall i$, G admits a T-SE.*

References

1. Andelman, N., Feldman, M., Mansour, Y.: Strong Price of Anarchy. In: SODA 2007 (2007)
2. Ashlagi, I., Krysta, P., Tennenholtz, M.: Social context games. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 675–683. Springer, Heidelberg (2008)
3. Aumann, R.: Acceptable Points in General Cooperative n-Person Games. Contributions to the Theory of Games 4 (1959)
4. Epstein, A., Feldman, M., Mansour, Y.: Strong Equilibrium in Cost Sharing Connection Games. In: ACMEC (2007)
5. Fiat, A., Kaplan, H., Levi, M., Olonetsky, S.: Strong price of anarchy for machine load balancing. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 583–594. Springer, Heidelberg (2007)
6. Fotakis, D., Kontogiannis, S., Spiraklis, P.: Atomic Congestion Games Among Coalitions. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 572–583. Springer, Heidelberg (2006)
7. Hayrapetyan, A., Tardos, E., Wexler, T.: The Effect of Collusion in Congestion Games. In: 38th ACM Symposium on Theory of Computing (2006)
8. Holzman, R., Law-Yone, N.: Strong equilibrium in congestion games. Games and Economic Behavior 21, 85–101 (1997)
9. Holzman, R., Law-Yone(Lev-tov), N.: Network structure and strong equilibrium in route selection games. Mathematical Social Sciences 46, 193–205 (2003)
10. Kuniavsky, S.: Coalitional Congestion Games. In: Technion. MSc Thesis (2007)
11. Leonardi, S., Sankowski, P.: Network Formation Games with Local Coalitions. In: PODC (2007)
12. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14, 124–143 (1996)
13. Borm, P., Facchini, G., van Megen, F., Tijs, S., Voorneveld., M.: Congestion games and potentials reconsidered. International Game Theory Review 46, 283–299 (1999)
14. Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. International Journal of Game Theory 2, 65–67 (1973)
15. Tennenholtz, M., Zohar, A.: Learning equilibrium in congestion games. In: AAMAS-2009 (2009)

Better with Byzantine: Manipulation-Optimal Mechanisms

Abraham Othman and Tuomas Sandholm

Computer Science Department
Carnegie Mellon University
{aothman,sandholm}@cs.cmu.edu

Abstract. A mechanism is manipulable if it is in some agents' best interest to misrepresent their private information. The *revelation principle* establishes that, roughly, anything that can be accomplished by a manipulable mechanism can also be accomplished with a truthful mechanism. Yet agents often fail to play their optimal manipulations due to computational limitations or various flavors of incompetence and cognitive biases. Thus, manipulable mechanisms in particular should anticipate byzantine play. We study *manipulation-optimal* mechanisms: mechanisms that are undominated by truthful mechanisms when agents act fully rationally, and do better than any truthful mechanism if *any* agent fails to act rationally *in any way*. This enables the mechanism designer to do better than the revelation principle would suggest, and obviates the need to predict byzantine agents' irrational behavior. We prove a host of possibility and impossibility results for the concept which have the impression of broadly limiting possibility. These results are largely in line with the revelation principle, although the considerations are more subtle and the impossibility not universal.

1 Introduction

Mechanism design is the science of generating rules of interaction—such as auctions and voting protocols—so that desirable outcomes result despite participating agents (humans, companies, software agents, etc.) acting in their own interests. A mechanism receives a set of preferences (i.e. type *reports*) from the agents, and based on that information imposes an *outcome* (such as a choice of president, an allocation of items, and potentially also payments).

A central concept in mechanism design is *truthfulness*, which means that an agent's best strategy is to report its type (private information) truthfully to the mechanism. The *revelation principle*, a foundational result in mechanism design, proves that any social choice function that can be implemented in some equilibrium form can also be implemented using a mechanism where all the agents are motivated to tell the truth. The proof is based on simply supplementing the manipulable mechanism with a strategy formulator for each agent that acts strategically on the agent's behalf (see, e.g., [1]). Since truthfulness is certainly worth something—simplicity, fairness, and the removal of incentives to invest in information gathering about others—the revelation principle produces something for

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 60–71, 2009.

nothing, a free lunch. As a result, mechanism design research has largely focused on truthful mechanisms.

In this work, we explore what can happen in manipulable mechanisms when agents do not play optimally. Is it possible to design mechanisms with desirable off-equilibrium properties? There are several reasons why agents may fail to play their optimal manipulations. Humans may play sub-optimally due to cognitive limitations and other forms of incompetence. The field of behavioral game theory studies the gap between game-theoretic rationality and human behavior (an overview is given in [2]). Agents may also be unable to find their optimal manipulations due to computational limits: finding an optimal report is NP-hard in many settings (e.g., [3,4,5,6]), and can be #P-hard [4], PSPACE-hard [4], or even uncomputable [7]. One notable caveat is that an agent's inability to find its optimal manipulation does not imply that the agent will act truthfully. Unable to solve the hard problem of finding its optimal manipulation, an agent may submit its true private type but she could also submit her best guess of what her optimal manipulation might be or, by similar logic, give an arbitrary report. A challenge in manipulable mechanisms is that it is difficult to predict in which specific ways agents will behave if they do not play according to game-theoretic rationality. Byzantine players, who behave arbitrarily, capture this idea.

In this paper, we explore mechanism design beyond the realm of truthful mechanisms using a concept we call *manipulation optimality*, where a mechanism benefits—and does better than any truthful mechanism—if *any* agent fails *in any way* to play her optimal manipulation. This enables the mechanism designer to do better than the revelation principle would suggest, and obviates the need to predict agents' irrational behavior. Conitzer and Sandholm [5] proved the existence of such a mechanism in one constructed game instance, but this work is the first to explore the concept formally and broadly.

2 The General Setting

Each agent i has type $\theta_i \in \Theta_i$ and a utility function $u_i^{\theta_i}(o) : O \rightarrow \Re$, which depends on the outcome $o \in O$ that the mechanism selects. An agent's type captures all of the agent's private information. For brevity, we sometimes write $u_i(o)$. A mechanism $M : \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n \rightarrow O$ selects an outcome based on the agents' type reports.

The mechanism designer has an objective (which can be thought of as mechanism utility) which maps outcomes to real values:

$$\mathcal{M}(o) = \sum_{i=1}^{n} \gamma_i u_i(o) + m(o),$$

where $m(\cdot)$ captures the designer's desires unrelated to the agents' utilities, and $\gamma_i \geq 0$. This formalism has three widely-explored objectives as special cases:

- Social welfare: $\gamma_i = 1$ and $m(\cdot) = 0$.
- Affine welfare: $\gamma_i > 0$ and $m(\cdot) \geq 0$.

– Revenue: Let outcome o correspond to agents' payments, $\pi_1(o), \ldots, \pi_n(o)$, to the mechanism. Fix $\gamma_i = 0$ and $m(o) = \sum_{i=1}^{n} \pi_i(o)$.

Definition 1. *Agent i has a* manipulable *type θ_i if, for some report of the other agents' types θ_{-i}, there exists $\theta_i' \neq \theta_i$ such that*

$$u_i(M(\theta_i', \theta_{-i})) > u_i(M(\theta_i, \theta_{-i}))$$

Note that a type that is manipulable for some reports of the other agents, but not for other reports of the other agents, is still manipulable.

Definition 2. *Types θ_i and θ_i' are* distinct *if there exists some report of other agents θ_{-i}, such that the best response for type θ_i is to submit t and the best response for type θ_i' is t', where*

$$M(t, \theta_{-i}) \equiv o \neq o' \equiv M(t', \theta_{-i}), \text{ and}$$

$$u_i^{\theta_i}(o) > u_i^{\theta_i}(o'), \text{ and } u_i^{\theta_i'}(o') > u_i^{\theta_i'}(o)$$

Put another way, types are distinct only if there exists a circumstance under which agents with those types will be motivated to behave distinctly, causing distinct outcomes that provide distinct payoffs.

Definition 3. *A mechanism is (dominant-strategy)* truthful *if no agent has a manipulable type.*

Definition 4. *Let f and g be functions mapping an arbitrary set $S \rightarrow \Re$. We say f* Pareto dominates *g (or g* is Pareto dominated by *f) if for all $s \in S$,*

$$f(s) \geq g(s),$$

where the inequality is strict for at least one s.

Definition 5. *A type report of $\theta_i^* \in \Theta_i$ is* optimal *for agent i if, given reports of other agents θ_{-i}, $u_i(M(\theta^*, \theta_{-i})) \geq u_i(M(\theta, \theta_{-i}))$, for all $\theta \in \Theta_i$.*

Now we are ready to introduce the main notion of this paper. We define a manipulable mechanism to be *manipulation optimal* if it does as well as the best truthful mechanism if agents play their optimal manipulations, and strictly better if any agent fails to do so in any way:

Definition 6. *For an arbitrary collection of types, let o represent the outcome that arises when all agents with manipulable types play optimally, and let $\hat{o}$ represent an outcome that can arise when some agents with manipulable types do not play optimally. We call a manipulable mechanism $\hat{M}$ a* strictly manipulation optimal mechanism (strict MOM) *if:*

1. *No truthful mechanism Pareto dominates agents playing optimally in $\hat{M}$. (Here the inputs are the true types of the agents and we measure based on the mechanism designer's objective.)*

2. For all $\hat{o}$*,* $\mathcal{M}(\hat{o}) > \mathcal{M}(o)$*, where* $\mathcal{M}(\cdot)$ *represents the designer's objective.*

If instead of the second condition holding strictly (i.e., for all $\hat{o}$), it holds with equality in some places and with strict inequality in others, we call $\hat{M}$ a *Pareto manipulation optimal mechanism (Pareto MOM).*

We assume that, if an agent's optimal play is to reveal its true type, then it will do so. The mechanism, for instance, can publish which types are truthful, and it can be expected that those agents will behave rationally. With software agents, such behavior can be hard-coded. However, our setting and results translate straightforwardly to a fully byzantine setting, where the behavior of every agent (regardless of the truthfulness of their type) is arbitrary. We discuss this setting at the conclusion of this section.

On the other hand, agents with manipulable types may not behave optimally; for instance, finding an optimal manipulation can be computationally intractable. It is important to note that we do *not* assume that an agent necessarily tells the truth if it fails to find its optimal manipulation. We require that our MOMs do well for *any* failure to manipulate optimally.

2.1 A Broad Impossibility Result for Strict MOMs

While Conitzer and Sandholm [5] showed that manipulation-optimal mechanisms do exist, the following result strongly curtails their existence.

Proposition 1. *No mechanism satisfies Characteristic 2 of Definition 6 if any agent has more than one distinct manipulable type.*

Proof. Suppose for contradiction that $\hat{M}$ is a manipulable mechanism satisfying Characteristic 2 such that agent i has two distinct manipulable types. Let the types be a and b, and let $\mathbf{x}$ represent the reports of other agents where they express its distinction, so that agent i of type a has best response a', and agent i of type b has best response b', and:

$$\hat{M}(a', \mathbf{x}) \neq \hat{M}(b', \mathbf{x})$$

We first define the following shorthand notation:

$$\sum(a') \equiv \sum_{j \neq i} \gamma_j u_j(\hat{M}(a', \mathbf{x})) + m(\hat{M}(a', \mathbf{x}))$$
$$\sum(b') \equiv \sum_{j \neq i} \gamma_j u_j(\hat{M}(b', \mathbf{x})) + m(\hat{M}(b', \mathbf{x}))$$

Because $\hat{M}$ satisfies the strict form of Characteristic 2, we get the following two inequalities on mechanism utilities—for agent i of type b and agent i of type a, respectively.

$$\gamma_i u_i^b(\hat{M}(b', \mathbf{x})) + \sum(b') < \gamma_i u_i^b(\hat{M}(a', \mathbf{x})) + \sum(a')$$
$$\gamma_i u_i^a(\hat{M}(a', \mathbf{x})) + \sum(a') < \gamma_i u_i^a(\hat{M}(b', \mathbf{x})) + \sum(b')$$

But because a' and b' are distinct, $u_i^a(\hat{M}(a',\mathbf{x})) > u_i^a(\hat{M}(b',\mathbf{x}))$ and $u_i^b(\hat{M}(b',\mathbf{x})) > u_i^b(\hat{M}(a',\mathbf{x}))$. Thus since $\gamma_i \geq 0$ we have

$$\gamma_i u_i^b(\hat{M}(a',\mathbf{x})) + \sum(a') \leq \gamma_i u_i^b(\hat{M}(b',\mathbf{x})) + \sum(a')$$
$$\gamma_i u_i^a(\hat{M}(b',\mathbf{x})) + \sum(b') \leq \gamma_i u_i^a(\hat{M}(a',\mathbf{x})) + \sum(b')$$

Combining the first lines of the above two equation blocks yields $\sum(b') < \sum(a')$, while combining the second lines yields $\sum(a') < \sum(b')$, a contradiction. □

This impossibility result is driven by the strict inequality in Characteristic 2 of Definition 6. In the next section, we consider what happens to this result when we loosen the strict inequality.

2.2 A Characterization of Pareto MOMs

Recall that the difference between the two MOM concepts was that strict MOMs require that the mechanism always do strictly better when an agent plays sub-optimally, while Pareto MOMs only require that the mechanism does not do worse and does strictly better at some point when an agent plays sub-optimally. It follows that possibility results for strict MOMs implies possibility for Pareto MOMs, and impossibility results for Pareto MOMs imply impossibility for strict MOMs.

We now revisit the impossibility of Proposition 1, but with the Pareto MOM notion. Instead of obtaining impossibility, we derive the following result:

Proposition 2. *In any mechanism that satisfies the Pareto version of Characteristic 2 of Definition 6, the report of every type that is a manipulable best-response for any other type results in identical mechanism utility. Furthermore, for any agent i with more than one distinct manipulable type,* $\gamma_i = 0$.

Proof. This proof follows the same guidelines as the one for strict MOMs. Define a, b, a', b', $\mathbf{x}$, $\sum(a')$ and $\sum(b')$ as before. The difference is that we now have only:

$$\gamma_i u_i^b(\hat{M}(b',\mathbf{x})) + \sum(b') \leq \gamma_i u_i^b(\hat{M}(a',\mathbf{x})) + \sum(a')$$
$$\gamma_i u_i^a(\hat{M}(a',\mathbf{x})) + \sum(a') \leq \gamma_i u_i^a(\hat{M}(b',\mathbf{x})) + \sum(b')$$

because there is no guarantee that the strict relation is expressed at $\mathbf{x}$. But because a and b are distinct, $u_i^a(\hat{M}(a',\mathbf{x})) > u_i^a(\hat{M}(b',\mathbf{x}))$ and $u_i^b(\hat{M}(b',\mathbf{x})) > u_i^b(\hat{M}(a',\mathbf{x}))$. Now if $\gamma_i > 0$, we have

$$\gamma_i u_i^b(\hat{M}(a',\mathbf{x})) + \sum(a') < \gamma_i u_i^b(\hat{M}(b',\mathbf{x})) + \sum(a')$$
$$\gamma_i u_i^a(\hat{M}(b',\mathbf{x})) + \sum(b') < \gamma_i u_i^a(\hat{M}(a',\mathbf{x})) + \sum(b')$$

which yields a contradiction. However, if $\gamma_i = 0$, we get

$$\gamma_i u_i^b(\hat{M}(a', \mathbf{x})) + \sum(a') = \gamma_i u_i^b(\hat{M}(b', \mathbf{x})) + \sum(a')$$
$$\gamma_i u_i^a(\hat{M}(b', \mathbf{x})) + \sum(b') = \gamma_i u_i^a(\hat{M}(a', \mathbf{x})) + \sum(b')$$

which, when combined with the MOM characterization above, yields possibility only when $\sum(a') = \sum(b')$, which, because $\gamma_i = 0$, indicates that mechanism utility is identical for reports of a' and b'. □

Corollary 1. *There exist no mechanisms with the social welfare maximization objective that satisfy the Pareto version of Characteristic 2 of Definition 6 if any agent has more than one distinct manipulable type.*

Corollary 2. *In any mechanism that satisfies the Pareto version of Characteristic 2 of Definition 6, the mechanism utility corresponding to reports of types that are not the best responses of some manipulable type must be at least the mechanism utility obtained from the best-response type reports, with at least one report inducing strictly greater mechanism utility.*

Proposition 3. *If every type is a manipulable best response for some other type, then there exist no mechanisms that satisfy the Pareto version of Characteristic 2 of Definition 6.*

Proof. Since every type is a manipulable best response for some other type, we have that every outcome must have identical mechanism utility. But then the "at least one strict" condition of Pareto dominance fails. □

From Proposition 3 we see that we get almost as broad impossibility for Pareto MOMs as we did for strict MOMs (Proposition 1).

We consider the strict MOM notion more compelling than the Pareto MOM notion for two reasons:

- Strict inequality is in line with prior work. It was the MOM notion used in the original paper by Conitzer and Sandholm [5] that proved that MOMs exist (although they did not call the mechanisms MOMs).
- The motivation of MOMs is to have a mechanism that does better when agents make mistakes—not to impose artificial caveats on the mechanism designer's utility function. Thus we consider the blanket impossibility result that we obtained for strict MOMs more relevant than the somewhat contrived, barely broader possibility we obtained for Pareto MOMs.

The results in this section extend straightforwardly to a fully byzantine setting, where all agents (including those with truthful types) behave arbitrarily. It is easy to see that no strict MOMs exist in this setting, because participating agents must have more than one type (or else the setting would not require the report of private information), and so the impossibility result of Proposition 1 holds. Furthermore, while there can exist Pareto MOMs for the fully byzantine setting, because truthful types are their own best response the results of Proposition 2 and its corollaries hold, in the sense that the report of any truthful type must also result in identical mechanism utility. Finally, for the fully byzantine setting, Proposition 3 adjusts so that if every type is the best response for *any* other type (rather than only just manipulable types) we get impossibility.

For the remainder of the paper, we return to our original setting, in which only players with manipulable types are byzantine. We feel the argument that truthful behavior for certain types can be hard-coded into computational agents, and publicly published and verified for human agents, to be the most convincing reason why we should expect players with truthful types to actually behave truthfully.

2.3 Single-Agent Settings

In this subsection we study settings where there is only one agent reporting its private information. If there are other agents, their types are assumed to be known, so there is only one *type-reporting* agent.

Proposition 4. *There exist no single-agent Pareto MOM with the objective of social welfare maximization.*

Proof. In the single-agent context, social welfare maximization indicates that the utility of the mechanism is equivalent to the utility of the single agent. Let the agent have manipulable type a, which has optimal report a'. Denote $\hat{a}$ as the report satisfying the strict Pareto MOM criterion (we could have $\hat{a} = a$, but both $a \neq a'$ (because a is manipulable) and $\hat{a} \neq a'$ hold). In particular:

$$u^a(\hat{M}(\hat{a})) > u^a(\hat{M}(a'))$$

but a' was an optimal report, so:

$$u^a(\hat{M}(a')) \geq u^a(\hat{M}(\hat{a}))$$

which is a contradiction. □

The impossibility for Pareto MOMs directly implies impossibility for strict MOMs.

Proposition 5. *There exist single-agent strict MOMs with the objective of affine welfare maximization.*

Proof. We can derive this result from the constructive proof of Conitzer and Sandholm [5] by recasting parts of their construction within our framework.

There exists a manager with three possible true types for a team of workers that needs to be assembled:

- "Team with no friends", which we abbreviate TNF.
- "Team with friends", which we abbreviate TF.
- "No team preference", which we abbreviate NT.

The mechanism implements one of two outcomes: picking a team with friends (TF), or picking a team without friends (TNF). The manager gets a base utility 1 if TNF is chosen, and 0 if TF is chosen. If a manager has a team preference, implementing that team preference (either with or without friends) gives the manager an additional utility of 3.

In addition to the manager, the other agent in the game is the HR director, who has utility 2 if a team with friends is chosen. Even though there are two agents in the game, because the HR director does not report a type, this is not a multiagent setting. In fact, the HR director's utilities are equivalent to the payoffs from the outcome-specific mechanism utility map $m(\cdot)$ (as we defined earlier in this paper).

The optimal truthful mechanism maps reports of NT and TNF to TNF and TF to TF. Now consider the manipulable mechanism that maps reports of TNF to TNF and NT and TF to TF. Note that in this mechanism there is only one manipulable type, NT, and that its optimal strategic play is to report TNF. This mechanism is manipulation-optimal: if the manager has type NT and reports NT or TF instead of TNF, the mechanism generates affine welfare of 2, whereas the optimal truthful mechanism generates affine welfare of 1. □

This possibility of strict MOMs implies possibility of Pareto MOMs.

In this example, it is NP-hard for an NT agent to report TNF because constructing a team of size k without friends requires solving the independent set problem in a graph of people where the edges are friend relationships [5]. Computational complexity is a strong justification for why an agent may not be able to find its optimal manipulation.

2.4 Multi-agent Settings

Though we proved above that there do not exist single-agent social welfare maximizing MOMs, they do exist in multi-agent settings!

Proposition 6. *There exist strict multi-agent MOMs with the objective of social welfare maximization.*

Proof. Consider a mechanism in which two agents, the row agent and the column agent, can have one of two types each, a or a'. Our mechanism maps reports to one of four different outcomes:

Report	a'	a
a'	o_1	o_2
a	o_3	o_4

The following two payoff matrices over the four outcomes constitute a manipulation-optimal mechanism. Payoffs for type a are on the left and payoffs for type a' are on the right:

Report	a'	a
a'	1,1	4,0
a	0,3	3,0

Report	a'	a
a'	3,4	5,0
a	0,6	0,0

Another way to view these payoffs is the following table:

Outcome	θ	u_{row}	u_{column}
o_1	a	1	1
	a'	3	4
o_2	a	4	0
	a'	5	0
o_3	a	0	3
	a'	0	6
o_4	a	3	0
	a'	0	0

In the mechanism, reporting a' is a strictly dominant strategy for agents of both types. By the revelation principle, we can "box" this mechanism into a truthful mechanism, M_1, that always chooses o_1. However, when an agent of type a plays a rather than a', social welfare is strictly higher than with o_1 (this property holds regardless of how the other agent behaves). We have now proven (the strict form of) Characteristic 2.

What remains to be proven is Characteristic 1: we need to prove that M_1 is Pareto undominated among truthful mechanisms. We begin by examining the following table, which shows the social welfare (sum of agents' utilities) for the four possible true type combinations (listed as $\theta_{row}, \theta_{column}$).

True types	o_1	o_2	o_3	o_4
a, a	2	4	3	3
a, a'	5	4	6	3
a', a	4	5	3	0
a', a'	7	5	6	0

Suppose that there exists a truthful mechanism, M^D, that Pareto dominates M_1. Note that M_1 delivers the highest payoff when both agents are of type a'. Thus, $M^D(a', a') = o_1$. But this implies that $M^D(a, a')$ and $M^D(a', a)$ must also equal o_1: mapping them to the outcome that gives higher social welfare (in the former case, o_3, and in the latter, o_2) is not truthful because the agent of type a has incentive to report a' and force o_1. At the same time, mapping to an outcome that is not o_1 delivers less social welfare than M_1. So, $M^D(a', a') = M^D(a', a) = M^D(a, a') = o_1$. But if these three inputs map to o_1, M^D cannot truthfully map revelations of (a, a) to any outcome other than o_1, because some agent will always want to deviate by reporting type a', and force outcome o_1. Therefore $M^D = M^1$ and so M^1 is undominated among truthful mechanisms.

□

The result above uses dominant strategies as the solution concept. Therefore, the result implies possibility for weaker equilibrium notions as well, such as Bayes-Nash equilibrium. Furthermore, this possibility for strict MOMs implies possibility for Pareto MOMs.

Definition 7. *An* anonymous *mechanism selects an outcome based only on the distribution of reported types, rather than based on the identities of the agents who reported those types.*

Definition 8. *Let i and j be any two* symmetric *agents, θ be a true type, $\widehat{\theta}$ be a report, and $\mathbf{x}$ be some report of the $n-1$ other agents. Then $u_i^{\theta}(M(\widehat{\theta}, \mathbf{x})) = u_j^{\theta}(M(\widehat{\theta}, \mathbf{x}))$ for all true types θ, all reports $\widehat{\theta}$ and all other report vectors $\mathbf{x}$.*

The agents in our construction in the proof above are not symmetric. We may ask whether MOMs exist for what can be considered the most common setting: where agents are symmetric, the equilibrium concept is dominant strategies, the mechanism is anonymous, and the objective is welfare maximization.

Proposition 7. *There exist no dominant-strategy anonymous strict multi-agent MOMs with the objective of social welfare maximization for symmetric agents.*

Proof. By Proposition 1, we can restrict attention to settings with a single manipulable type. Call the type a, and let the best report of that type of an agent be a'. Suppose mechanism $\hat{M}$ satisfies Characteristic 2. By the revelation principle it has a corresponding truthful mechanism M. We show that we can construct a truthful mechanism M^D that Pareto dominates M.

First, if a set of reports includes a type other than a or a', we set M^D to simply mirror the action taken by M. Strategic implications for agents other than types a and a' are unaffected because for agents of those types, reporting the true type was a dominant strategy under $\hat{M}$.

Let o be the outcome implemented by M when all agents report a, and let o' be the outcome implemented by M when all agents report a'. Denote by $\tilde{a}$ any combination of reports a and a'; observe that $M(\tilde{a}) = o'$.

By Characteristic 2 we know that we get higher social welfare if agents of type a—whose best manipulation is to report a'—cannot find the manipulation and report a instead. Since agents are symmetric, this implies $u^a(o') < u^a(o)$. This is akin to the Prisoner's Dilemma: the dominant strategy of type a is to report a', but the outcome is worse for agents if they all report a' rather than a.

Now we construct M^D based on the payoff structure of agents of type a'.

- **Case I:** $u^{a'}(o') < u^{a'}(o)$. In this case we let M^D map each $\tilde{a}$ to o. M^D Pareto dominates M.
- **Case II:** $u^{a'}(o') \geq u^{a'}(o)$. In this case we let M^D select o if all agents report a, and o' for any other $\tilde{a}$. M^D Pareto dominates M. Note that M^D is identical to M for all reports except the one where all agents report a.

While the impossibility results earlier in this paper were based on a violation of Characteristic 2 of MOMs alone, here the impossibility comes from not being able to satisfy Characteristics 1 and 2 together. □

We use the strict MOM concept here rather than the Pareto MOM concept, because we cannot assert that $u^a(o') < u^a(o)$ necessarily in the Pareto context. Both our possibility results and this impossibility result have used the dominant strategy solution concept. This implies the strongest possibility, but the weakest impossibility. Here, our requirement for dominant strategy manipulability avoids issues with degenerate special cases.

We can circumvent the above impossibility by moving to the affine welfare objective. Note that for an anonymous mechanism, the outcome-specific mechanism utility function $m(\cdot)$ can depend only on the distribution of types, rather than the identities of the agents reporting those types.

Proposition 8. *There exist dominant-strategy anonymous multi-agent strict MOMs with the objective of affine welfare maximization, even for symmetric agents.*

Proof. We provide a constructive proof with the same structure as Proposition 6, but now let the payoff matrices be as follows (the left matrix is for type a and the right matrix for type a').

Report	a'	a
a'	2,2	1,1
a	1,1	0,0

Report	a'	a
a'	4,4	1,3
a	3,1	0,0

Let $\gamma_i = 1$ for all i, and let the mechanism's additional payoff, $m(\cdot)$, be $\{0, 3, 3, 5\}$ for outcomes o_1 through o_4, respectively. Note that the row and column agents are symmetric (the payoff matrices are symmetric) and that $m(o_2) = m(o_3)$. The dominant strategy is for every agent to report type a'. Therefore this mechanism has truthful analogue M_1, the mechanism that always chooses o_1.

We now show that M_1 is Pareto undominated among truthful mechanisms. First, note that M_1 maximizes the objective when both agents have type a'. It can be shown that (using a construction akin to the last table in the proof of Proposition 6) that due to agent incentives to deviate, any truthful mechanism that would dominate M_1 must map all reports to o_1. Thus M_1 is Pareto undominated among truthful mechanisms.

The manipulation-optimality of the mechanism defined by the payoff matrices above comes from noting that whenever agents of type a fail to report a', affine welfare is strictly higher. □

3 Conclusions and Future Work

The strategic equivalence of manipulable and non-manipulable mechanisms—captured by the revelation principle—does not mean that every manipulable mechanism is automatically flawed. It is well-known that agents often fail to play their optimal manipulations in mechanisms due to computational limitations or various flavors of incompetence and cognitive biases. Yet it is difficult to predict how such game-theoretically irrational agents will act (or which particular equilibrium, among many, each agent will play). We studied the notion of *manipulation-optimal mechanisms*: mechanisms that are undominated by truthful mechanisms when agents play fully rationally, and do better than any truthful mechanism if *any* agent fails to play rationally *in any way.* This enables the mechanism designer to do better than the revelation principle would suggest, and obviates the need to predict agents' irrational behavior.

For the general setting, we showed that manipulation optimality is limited to mechanisms that have at most one manipulable type per agent. We also proved a host of other impossibility and possibility results for the existence of manipulation-optimal mechanisms for a variety of settings and mechanism design objectives. In particular, the possibility result for strict MOMs in the multi-agent social welfare maximization setting was very surprising. However, the overall impression was one of broad impossibility. Thus, our results suggest that in many settings there is a "cost of manipulability": implementing a manipulable mechanism inherently exposes the designer to achieving an unnecessarily poor result when agents do not perform optimally.

Manipulation-optimal mechanisms open an avenue for numerous forms of future research. For one, it would be interesting to study manipulation optimality under other objectives, such as notions of fairness. As another direction, we plan to explore whether *automated mechanism design* [8] can be used to design manipulation-optimal mechanisms. Given priors over types (and perhaps also over behaviors), it may be possible to ignore incentive compatibility constraints and design manipulable mechanisms that yield higher mechanism utility.

Acknowledgements

This material is based upon work supported by the National Science Foundation under ITR grant IIS-0427858. An earlier version of this work appeared in COMSOC-08.

References

1. Mas-Colell, A., Whinston, M., Green, J.R.: Microeconomic Theory. Oxford University Press, New York (1995)
2. Camerer, C.: Behavioral Game Theory: Experiments in Strategic Interaction. Princeton University Press, Princeton (2003)
3. Bartholdi III, J., Tovey, C., Trick, M.: The computational difficulty of manipulating an election. Social Choice and Welfare 6(3), 227–241 (1989)
4. Conitzer, V., Sandholm, T.: Universal voting protocol tweaks to make manipulation hard. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, pp. 781–788 (2003)
5. Conitzer, V., Sandholm, T.: Computational criticisms of the revelation principle. In: The Conference on Logic and the Foundations of Game and Decision Theory (LOFT), Leipzig, Germany (2004); Earlier versions: AMEC-2003, EC-2004
6. Procaccia, A.D., Rosenschein, J.S.: Junta Distributions and the Average-Case Complexity of Manipulating Elections. Journal of Artificial Intelligence Research (JAIR) 28, 157–181 (2007)
7. Nachbar, J.H., Zame, W.R.: Non–computable strategies and discounted repeated games. Economic Theory 8(1), 103–122 (1996)
8. Conitzer, V., Sandholm, T.: Complexity of mechanism design. In: Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI), Edmonton, Canada, pp. 103–110 (2002)

On the Planner's Loss Due to Lack of Information in Bayesian Mechanism Design*

José R. Correa and Nicolás Figueroa

CEA, Departamento de Ingeniería Industrial, Universidad de Chile
{jcorrea,nicolasf}@dii.uchile.cl

Abstract. In this paper we study a large class of resource allocation problems with an important complication, the utilization cost of a given resource is private information of a profit maximizing agent. After reviewing the characterization of the optimal bayesian mechanism, we study the informational cost introduced by the presence of private information. Our main result is to provide an upper bound for the ratio between the cost under asymmetric information and the cost of a fully informed designer, which is independent of the combinatorial nature of the problem and only depend on the statistical distribution of the resource costs. In particular our bounds evaluates to 2 when the utilization cost's distributions are symmetric and unimodal and this is tight. We also show that this bound holds for a variation of the Vickrey-Clark-Groves mechanism, which always achieves an ex-post efficient allocation. Finally we point out implementation issues of the considered mechanisms.

Keywords: Mechanism Design, Information Cost.

JEL Classification: C60, C72, D44.

1 Introduction

A wide class of problems of the form $\min\{c^t x | x \in \Gamma\}$ have been analyzed in the literature and their applications to real world problems are vast. In this paper, we consider such a class of problems with an important and realistic complication, the utilization cost of a given resource x_i is private information of a profit maximizing agent.

For example, let us consider a natural situation in supply chain management. A large company needs to procure quantities D_i of a given good for its various locations $t_1, \dots, t_k$. The good is produced at various locations $s_1 \dots, s_l$, each of them with a maximum production capacity Q_j. The delivery of the goods is done through a transportation network in which each link has a cost that is publicly known. This problem, when the production facilities are owned by the company, reduces to a standard minimum cost flow through a network. If, however, the production facilities are owned by private contractors, whose production cost is private information, there is an added layer of difficulty to the problem. Now the company must design a mechanism to minimize

* This work was partially supported by FONDECYT through grants 11070159 and 1090050 and by the Instituto Milenio en Sistemas Complejos de Ingeniería.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 72–84, 2009.

expected procurement cost, subject to the feasibility constraints on the network, and inducing contractors to reveal their costs in exchange of a profit.

The main contribution of this paper is to study the informational cost introduced by the presence of private information. Such a consideration is important, because it lies at the heart of an old economic question: To make or to buy? If this cost is small, the organizational cost of acquiring small producers may be high and not worth it. If big, such an acquisition may turn out to be profitable for the company trying to procure goods or services. Our main result is to provide an upper bound for the ratio between the cost under asymmetric information and the cost of a fully informed designer. Specifically, we show that for a large class of distributions, containing those that are symmetric and unimodal, the expected cost of an optimal mechanism is at most twice the cost of an optimal solution obtained by a fully informed planner. Neither this bound nor its tightness depend on the combinatorial nature of the problem, but only on the statistical distribution of private information. The latter bound holds for a variation of the Vickrey-Clark-Groves (VCG) mechanism as well, and becomes significantly better in some special situations.

Related questions were studied by Bulow and Klemperer [2], who analyze the suboptimality, in terms of revenue, of VCG for a single unit auction. They show that one extra bidder in a VCG format gives more revenue than the Myerson mechanism. Recently, Aggarwal et al. [1] study the suboptimality, in terms of efficiency, of the Myerson auction, showing that $\Theta(\log k)$ extra bidders suffice to match the efficiency of a VCG mechanism with k bidders, and generalize the result to multiunit auctions. Also, Elkind et al. [4] establish bounds on the payments of the VCG and optimal mechanisms in path auctions, and point out that these may differ significantly. Finally, Hartline and Roughgarden [5] consider similar issues in the context of money burning mechanisms.

We also study the computational cost of calculating an optimal mechanism. We show that such a problem is equivalent to performing parametric linear programming over the set Γ, which is in general of exponential complexity, even if optimization over Γ is simple. For the important class of problems where the set Γ is a 0-1 polytope, however, we give a simple algorithm with the same complexity of the original optimization problem with complete information. For the other problems, we point out that a simple sampling technique, which takes advantage of the owners' risk neutrality, gives a random mechanism yielding the same expected cost as the deterministic one.

The paper is organized as follows. In section 2 we quickly review the characterization of the optimal bayesian (i.e., utilization costs are random variables) mechanism for the whole class of problems with a linear cost function and a fixed constraint set. Our main results concerning the informational cost are found in section 3, while the computational considerations are discussed in section 4.

2 The Model

2.1 The Environment

We consider a setting in which scarce resources must be allocated to carry out a given project. The cost of each resource may be public or private information. Depending on the situation, the planner's goal is to minimize her own expected cost, or the social cost of the project. To this end, she can design a mechanism where the owners with private information have incentives to reveal their private information.

In our framework each resource $a \in A$ is represented by a variable x_a, and is associated with a marginal cost of utilization c_a. The set A is partitioned into two sets A_1 and A_2. Costs of resources $a \in A_1$ are private, and thus c_a is private information and is distributed according to F_a, whose bounded support is the interval $[\underline{c}_a, \overline{c}_a] \subset \mathbb{R}_+$. The distribution F_a is assumed to have a density f_a which is continuous and strictly positive in $[\underline{c}_a, \overline{c}_a]$. For simplicity we also assume that $F_a(c_a)/f_a(c_a)$ is nondecreasing (satisfied among others by the family of logconcave distributions). Costs of resources $a \in A_2$ are public information and equal c_a. Resources are scarce and subject to an exogenous feasibility constraint $x \in \Gamma \subseteq \mathbb{R}^{|A|}$, which we assume compact. Therefore, if all costs c_a, $a \in A$ were known, the planner would solve $\min\{c^T x : x \in \Gamma\}$. However, costs of resources in A_1 are unknown and thus the planner must design a mechanism to elicit this information in order to achieve her goal.

We now give a key property that holds in this environment. It states that if the cost of a resource increases, the value of the corresponding variable, in a cost-minimizing solution, does not increase. This intuitive and simple result turns out to be critical for characterizing the optimal mechanism.

Lemma 1. *Let $x(c) = \{x_a(c)\}_{a \in A}$ be the minimum cost assignment in Γ for a cost vector c. Then $x_a(\cdot, c_{-a})$ is non-increasing for all $a \in A$.*

Proof. Consider a cost vector c and let c' be defined as $c'_e = c_e$ for all $e \in A - \{a\}$ and $c'_a = c_a + \varepsilon$ for some $\varepsilon > 0$. From the definition of $x(c)$ we have that: $c^T x(c) \le c^T x(c')$ and ${c'}^T x(c') \le {c'}^T x(c)$. Summing both terms we obtain $(c^T - {c'}^T)[x(c) - x(c')] \le 0$ which is equivalent to $x_a(c) \le x_a(c')$. □

As usual, if $x \in \mathbb{R}^n$, x_{-i} denotes the vector in which the $i-$th component is removed. We also define: $f(c) = \prod_{a \in A_1} f_a(c)$, $f_{-a}(c) = \prod_{e \in A_1 - \{a\}} f_a(c)$, $C = \prod\limits_{a \in A_1} [\underline{c}_a, \overline{c}_a]$, $C_{-a} = \prod_{e \in A_1 - \{a\}} [\underline{c}_a, \overline{c}_a]$.

2.2 Mechanisms

In order to achieve her objectives, the planner designs a mechanism. In other words, the planner chooses a message space M_a for each $a \in A_1$, together with assignment and payment rules. Given messages from resource owners, these rules determine the amount of each resource used by the planner and the payment received by each owner. Due to the revelation principle it is enough to set $M_a = [\underline{c}_a, \overline{c}_a]$ and consider truthful mechanisms.

Therefore, a mechanism is given by assignment rules $\{x_a\}_{a \in A}$, indicating how much of resource a will be used, and a family of payment rules $\{t_a\}_{a \in A_1}$, indicating the total payment to the owner of resource $a \in A_1$. Naturally, these values depend on the cost revelations of each owner, therefore $x_a : C \longrightarrow \mathbb{R}$ and $t_a : C \longrightarrow \mathbb{R}$. Our framework allows the payment received by the owner of resource a, given revelations c, to be random. If this is the case, t_a denotes the total expected payment to the owner of resource $a \in A_1$. The payoff of the owner of resource a, with cost c_a, when reporting a cost c'_a is given by:

$$U_a(c_a, c'_a) = \int_{C_{-a}} [t_a(c'_a, c_{-a}) - c_a x_a(c'_a, c_{-a})] f_{-a}(c_{-a}) dc_{-a}. \tag{1}$$

The payoff of a resource owner with cost c_a is then:

$$V_a(c_a) = \max_{c'_a \in C_a} U_a(c_a, c'_a). \tag{2}$$

We must also consider mechanisms that give a positive utility to owners and satisfy the feasibility constraints. We can thus give the following definition.

Definition 1. *A mechanism $(x, t) \equiv (\{x_a\}_{a \in A}, \{t_a\}_{a \in A_1})$ is feasible if and only if for all cost realizations c the following hold:*

$$\begin{aligned} &\text{(IC)}\ \ V_a(c_a) = U_a(c_a, c_a)\ \textit{for all } a \in A_1, \\ &\text{(PC)}\ V_a(c_a) \geq 0 \qquad\qquad \textit{for all } a \in A_1, \\ &\text{(F)}\quad x(c) \in \Gamma. \end{aligned}$$

The Optimal Bayesian Mechanism. With the previous definition, we can write the problem of a cost-minimizing designer as

$$\min \left\{ \int_{c \in C} \left(\sum_{a \in A_1} t_a(c) + \sum_{a \in A_2} c_a x_a(c) \right) f(c) dc \ :\ (x, t) \text{ is feasible} \right\}. \tag{3}$$

Using by now standard arguments introduced by Myerson [9] and extended among others by Elkind et. al. [4] (see [10, Chapter 13] for a detailed treatment) we can characterize the optimal Bayesian mechanism relying on Lemma 1. A proof can be found in the full version of this paper. Indeed, the optimal mechanism can be written as the solution to the following control problem

$$\min_{\{x_a(c)\}_{a \in A}} \int_{c \in C} \left(\sum_{a \in A_1} x_a(c) \left[c_a + \frac{F_a(c_a)}{f_a(c_a)} \right] + \sum_{a \in A_2} x_a(c) c_a \right) f(c) dc$$

$$\text{s.t.} \qquad x(c) \in \Gamma \text{ and } \nu_a(c_a) \text{ non-increasing for all } a \in A_1.$$

Here, $\nu_a(c_a) := \int_{C_{-a}} x_a(c_a, c_{-a}) f_{-a}(c_{-a}) dc_{-a}$ is the expected utilization of resource a for $a \in A_1$.

Because of Lemma 1, and the assumption that $F_a(c_a)/f_a(c_a)$ is increasing, we can relax the constraint asking for $\nu_a(c_a)$ non-increasing and solve the above problem pointwise to obtain a feasible solution. Therefore, we can characterize the optimal mechanism.

Proposition 1. *The optimal assignment rules* $\bar{x}(c) = \{\bar{x}_a(c)\}_{a \in A}$ *are those solving, for each cost revelations* $\{c_a\}_{a \in A_1}$*, the following optimization problem:*

$$\min_{y \in \Gamma} \sum_{a \in A_1} \left(c_a + \frac{F_a(c_a)}{f_a(c_a)} \right) y_a + \sum_{a \in A_2} c_a y_a,$$

and an optimal payment rule is given by $\bar{t}_a(c) = c_a \bar{x}_a(c) + \int_{c_a}^{\bar{c}_a} \bar{x}_a(t, c_{-a}) dt$*. In other words,* $x(c)$ *is the minimum cost assignment in* Γ *with virtual costs* $c'_a = c_a + F_a(c_a)/f_a(c_a)$ *for all* $a \in A_1$*, and* $c'_a = c_a$ *for all* $a \in A_2$*.*

The Truncated Vickrey-Clark-Groves Mechanism. On the other hand, a planner interested in achieving ex-post efficiency may consider the standard VCG mechanism, which solves, for every cost realization c, the problem $\min\{c^t x | x \in \Gamma\}$ and assigns according to the solution rule $x_a^V(c)$: It pays agent $a \in A_1$, $t_a(c) = c_a x_a(c) + (\sum_{b \in A} c_b x_b^{-a}(c) - \sum_{b \in A} c_b x_b(c))$, where $x^{-a}(c)$ is a solution of $\min\{c^t x | x \in \Gamma, x_a = 0\}$. It is well known that such a mechanism is incentive compatible, but can involve infinite costs. However, if the support of the cost distribution is known, payments can be bounded without losing incentive compatibility (and thus efficiency). We denote such a mechanism, with payments given by $t_a(c) = \min\{c_a x_a(c) + (\sum_{b \in A} c_b x_b^{-a}(c) - \sum_{b \in A} c_b x_b(c)), \bar{c}_a x_a(c)\}$, the Truncated Vickrey-Clark-Groves (TVCG) mechanism.

3 Loss Due to Lack of Information

The presence of private information among resource owners increases the cost of performing a given task. A natural problem, with relevant practical implications, is to quantify the relationship between the cost under complete and incomplete information. The former corresponds to a situation where the planner owns the different resources and the technology needed for their production, therefore knowing exactly the production costs. The latter corresponds to a decentralized situation, where the planner has outsourced the production of necessary inputs, and therefore does not know precisely their production costs. Since outsourcing can imply important savings in terms of managerial effort, it is critical to know how much is a firm losing by spinning off some of its components, or how much is a central planner losing by privatizing some key components of a planned economy. Moreover, with incomplete information, a cost-minimizing planner does not necessarily assign resources efficiently (since he considers modified costs), so we consider the question of the expected cost of an efficient mechanism, the TVCG, and its comparison to the cost-minimizing one and the fully informed solution.

Interestingly, both comparisons can be done independently of the combinatorial structure of the problem (given by the set Γ), and depend only on the nature of the incomplete information (given by the distribution functions F_a). The critical lemma is the following:

Lemma 2. *If the distribution* F*, with* $F(a) = 0$ *and density* f*, satisfies that* $\mathbb{E}(X | X \leq y) \geq y/\alpha$*, where* X *is drawn according to* F*, then for* $[a, b] \subset \mathbb{R}_+$ *and* $g(\cdot)$ *a nonnegative, non-increasing real-valued function defined on* $[a, b]$ *we have:*

$$\int_a^b g(c)F(c)dc \le (\alpha-1)\int_a^b g(c)cf(c)dc.$$

Proof. Let $g(\cdot)$ be any nonnegative non-increasing real-valued function and F be a distribution, with density f, satisfying the conditions in the proposition. Note that as $g(\cdot)$ is monotone, it is differentiable almost everywhere [7], thus $g'(c) \le 0$ a.e., implying that

$$\int_a^b g(c)(F(c)-(\alpha-1)cf(c))dc = g(b)\int_a^b (F(s)-(\alpha-1)sf(s))ds$$
$$-\int_a^b g'(c)\int_a^c (F(s)-(\alpha-1)sf(s))dsdc,$$

is nonpositive if $\int_a^y F(c)dc \le (\alpha-1)\int_a^y cf(c)dc$ holds for all $y \in [a,b]$. This latter condition is equivalent to $\mathbb{E}(X|\, X \le y) \ge y/\alpha$, since integrating by parts

$$\int_a^y F(c)dc - (\alpha-1)\int_a^y cf(c)dc = yF(y) - \alpha\int_a^y cf(c)dc,$$

which is nonpositive so long as $\mathbb{E}(X|\, X \le y) \ge y/\alpha$. □

3.1 Cost Loss Due to Lack of Information

We now turn compare the planner's expected cost when using the cost-minimizing and the TVCG mechanisms to that in case she had complete information. From the description in Section 2.2 (see full version for details), and noting that the worst type $\bar{c}_a$ gets 0 rents in both the cost-minimizing and the TVCG mechanism, we can write the expected cost of both mechanisms as:

$$\mathcal{C}_I = \min_{x(c)\in\Gamma}\int_{c\in C}\left(\sum_{a\in A_1} x_a(c)\left[c_a + \frac{F_a(c_a)}{f_a(c_a)}\right] + \sum_{a\in A_2} x_a(c)c_a\right) f(c)dc. \quad (4)$$

$$\mathcal{C}_{VCG} = \int\left[\sum_{a\in A_1} x_a^V(c)\left[c_a + \frac{F_a(c_a)}{f_a(c_a)}\right] + \sum_{a\in A_2} x_a^V(c)c_a\right] f(c)dc, \quad (5)$$

On the other hand, when complete information is available to the planner, her cost is given by:

$$\mathcal{C}_C = \min_{x(c)\in\Gamma}\sum_{a\in A_1}\int_{c\in C} c_a x_a(c)f(c)dc + \sum_{a\in A_2}\int_{c\in C} c_a x_a(c)f(c)dc. \quad (6)$$

Observe that if $A_1 = A$, that is all costs are private information, and F_a is uniform in $[0,s]$ for all $a \in A$, the planner's problem given by (4) is exactly the same as that in (6) with the costs doubled. Therefore the planner's expected cost in the optimal mechanism is twice as much as that in the complete information setting. Moreover, since the assignment rules of TVCG coincide with the fully informed solution, in this setting the cost of the TVCG mechanism is also twice $\mathcal{C}_I$. In what follows, we extend this result to a very general class of distribution functions, and prove that such a bound is also true for the comparison between the TVCG (which in general has a higher cost than $\mathcal{C}_C$) and the complete information mechanism.

With Lemma 2 at hand, the proof of the next result becomes remarkably simple. Its full significance though, will be evident in the next section, once we establish that large and natural classes of distributions satisfy the hypothesis.

Proposition 2. *If for all $a \in A_1$ the distribution F_a satisfies that $\mathbb{E}(X|\, X \leq y) \geq y/\alpha$, where X is drawn according to F_a, then $\mathcal{C}_I \leq \mathcal{C}_{VCG} \leq \alpha \cdot \mathcal{C}_C \leq \alpha \cdot \mathcal{C}_I$.*

Proof. The first and last inequalities are direct since we first compare the optimal mechanism to TVCG, and the fully informed optimal solution to an optimal mechanism. For the second one, we apply Lemma 2 to expression (5). Note that Lemma 2 holds even if the function $g(\cdot)$ is not continuous, as it may be the case for $x^V(\cdot)$, for instance, when the underlying set Γ is polyhedral or discrete. Thus we can write:

$$\begin{aligned}\mathcal{C}_{VCG} &= \int \left[\sum_{a \in A_1} x_a^V(c) \left[c_a + \frac{F_a(c_a)}{f_a(c_a)} \right] + \sum_{a \in A_2} x_a^V(c) c_a \right] f(c) dc \\ &\leq \alpha \int \sum_{a \in A} x_a^V(c) c_a f(c) dc = \alpha \mathcal{C}_C.\end{aligned}$$

The last equality holds since TVCG assigns efficiently. □

Note that the previous bound is related *only* to the distribution of private information about costs, and not to the particular problem Γ being considered. As we already pointed out, in *any* instance of a combinatorial problem defined by Γ, when all resources are private and the information is distributed uniformly on $[0, a]$, this bound is tight, since for a $\mathbb{E}(X|\, X \leq y) = y/2$.

Observation. A natural question is whether there is a better bound for the comparison between $\mathcal{C}_I$ and $\mathcal{C}_{VCG}$ than just $\mathcal{C}_{VCG} \leq \alpha \cdot \mathcal{C}_I$. If such a bound holds true when we consider the full information mechanism, is it possible to do better when considering the optimal mechanism under incomplete information? The answer is no, as sometimes the incomplete information planner has the same cost as the fully informed planner, while the TVCG mechanism performs badly at a cost $\alpha \cdot \mathcal{C}_C$. Consider for example the case where the planner must send one unit of flow between two nodes, in a two link network. One of the links is public while the other is private, i.e., $A_1 = \{a_1\}$, $A_2 = \{a_2\}$, and $\Gamma = \{(x_{a_1}, x_{a_2}) \geq 0 : \; x_{a_1} + x_{a_2} = 1\}$. Consider $\{F_{a_1}^{(n)}\}$ a family of symmetric and unimodal distributions for resource a_1, and assume that their support is the full interval $[0, 1]$ and that $F_{a_1}^{(n)} \longrightarrow \delta_{1/2}$, where $\delta_{1/2}$ is the mass distribution putting probability 1 to $c_{a_1} = 1/2$. Assume also that $c_{a_2} = 1$. Then, we have that $\mathcal{C}_{VCG}^{(n)} = 1$, but $\mathcal{C}_I^{(n)} \longrightarrow \mathcal{C}_C \equiv \frac{1}{2}$, while the value of α for these distributions is, as we will see next, 2. Therefore our bound is tight.

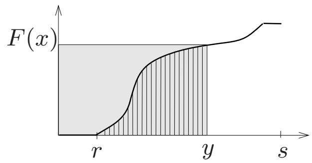


Fig. 1. For $\alpha = 2$ the condition of Proposition 2 states that the area under the curve is at most half of the gray area

3.2 Distributions

Having established that Proposition 2 holds independently of the combinatorial structure of the problem, the main question is thus to determine the distributions satisfying the hypothesis, and how small their corresponding value of α is. Note first that the proposition can be applied to densities which are non-decreasing with $\alpha = 2$. Therefore, for situations where agents are concentrated among "bad" providers, we can do as well as in the case with a uniform distribution.

Let us give a geometric interpretation of the inequality $\mathbb{E}(X | X \leq y) \geq y/\alpha$, where X is a random variable drawn according to a distribution F defined in an interval $[r, s]$. Writing down the expression and integrating by parts we note that the condition is equivalent to

$$(\alpha - 1)yF(y) \geq \alpha \int_r^y F(x)dx. \tag{7}$$

Thus the condition states that for any y in $[r, s]$ the area defined by the rectangle of width y and height $F(y)$ is at least a fraction $\alpha/(\alpha - 1)$ of the area comprised under the curve $F(x)$ between r and y. Figure 1 depicts the situation.

With the intuition provided by the above interpretation we are able to find a number of distributions for which Proposition 2 can be applied. A particularly relevant example occurs when the distribution which is a minimum between m draws of a uniform distribution. Here, we capture a situation where providers have a try at m different technologies and select the best of them. Such an environment is biased towards "good" providers through "natural selection", but even in this case we can provide a tight upper bound.

Proposition 3. *Consider agents whose cost is given by the minimum of m draws from a uniform distribution in $[0, 1]$. Their cost distribution is then given by $F(x) = 1 - (1 - x)^m$, and it satisfies $E(X|X \leq Y) \geq Y/(m + 1)$.*

Proof. Note first that, using condition 7, the inequality $E(X|X \leq y) \geq \frac{y}{m+1}$ is equivalent to

$$\max_{y\in[0,1]} \frac{\int_0^y (1 - (1 - s)^m)ds}{y(1 - (1 - y)^m)} \leq \frac{m}{m + 1},$$

which in turn can be rewritten as $y + (1 - y)^{m+1} + my(1 - y)^m \leq 1$ for all $y \in [0, 1]$. Further cancelations lead to $(1 - y)^m + my(1 - y)^{m-1} \leq 1$ for all $y \in [0, 1]$, and using the change of variables $s = 1 - y$, we obtain that $G(s) = s^m + mys^{m-1} \leq 1$ for all $s \in$

$[0,1]$. Noting that $G'(s) = ms^{m-1} + m(m-1)s^{m-2} - m^2 s^{m-1} = 0$ implies $s = 1$, and that $G(0) = 0$, $G(1) = 1$, the result follows. □

Our result, as we show next, can also be applied to an important class of symmetric distributions, which in particular includes those that are symmetric and unimodal (SUD). A distribution function is unimodal if it has a unique local maximum.

Proposition 4. *Suppose that for all $a \in A_1$ the distribution F_a with density f_a has support $[0,1]$, is symmetric, and satisfies $F_a(y) \leq y f_a(y)$ for $0 \leq y \leq 1/2$. Then we have that $\mathcal{C}_I \leq \mathcal{C}_{VCG} \leq 2 \cdot \mathcal{C}_C$.*

Proof. Because of Proposition 2 we just need to show that if X is a random variable drawn from a symmetric distribution F, whose density f has support $[0,1]$, and satisfies $F(y) \leq y f(y)$ for $0 \leq y \leq 1/2$, then $E(X|X \leq y) \geq y/2$. Using condition 7 this is equivalent to showing

$$yF(y) \geq 2 \int_0^y F(x)dx \quad \text{for all } y \in [0,1].$$

Note first that the inequality holds for $y = 0$. Furthermore, we know that for all $0 \leq y \leq 1/2$ we have $2F(y) \leq (F(y) + yf(y))$. The latter is equivalent to saying that the derivative of the left hand side of the condition above is larger than the derivative of the right hand side. Thus the condition holds for all $y \in [0, 1/2]$.

Furthermore, note that $F(y) \leq y$ for all $y \in [0, 1/2]$. Indeed, by contradiction assume that $F(z) > z$ for some $z \in [0, 1/2]$. In this case let $1/2 \geq z' > z$ be a real for which $F(z') \geq z'$ and such that $F'(z') = f(z') < 1$ (which has to exist since $F(1/2) = 1/2$). Now $F(z') \geq z' > z' f(z')$ which is a contradiction. Using the symmetry of F, this implies that $F(y) \geq y$ for all $y \in [1/2, 1]$.

Now, let $y \in [1/2, 1]$ and observe that

$$\int_0^y F(x)dx = \int_0^{1-y} F(x)dx + \int_{1/2-(y-1/2)}^{1/2+(y-1/2)} F(x)dx.$$

Using that $F(x) \leq x$ for $0 \leq x \leq 1/2$ to bound the first term and the symmetry of F to evaluate the second, we can write:

$$\int_0^y F(x)dx \leq \int_0^{1-y} xdx + \frac{2y-1}{2} \leq \frac{(1-y)^2}{2} + \frac{2y-1}{2} \leq y^2/2.$$

Using again that $F(y) \geq y$ for all $y \in [1/2, 1]$ we conclude that $y^2/2 \leq yF(y)/2$, which completes the proof. □

It is straightforward to extend the previous proposition to the case when the support of the distributions F_a is an interval $[r, s]$ with $r \geq 0$ and still satisfy the conditions of the proposition. If this is the case the bound becomes $\mathcal{C}_I \leq \frac{2s}{r+s} \cdot \mathcal{C}_C$. The intuition behind this result is natural. For instance, if r is very close to s, the cost under incomplete information approaches that of a fully informed planner. Also, if $s = r + K$, for constant K, the bound also goes to one as r goes to infinity. This is because the amount of information the planner ignores is irrelevant when compared to the total cost of the project.

Furthermore, if the distributions F_a for all $a \in A_1$ are symmetrical and unimodal, then f_a is nondecreasing in the interval $[0, 1/2]$. This implies that $F_a(y) \leq y f_a(y)$ for all $y \in [0, 1/2]$. Thus we have the following corollary.

Corollary 1. *If F_a is SUD for all $a \in A_1$, then $\mathcal{C}_I \leq 2 \cdot \mathcal{C}_C$. Moreover, if F_a is SUD on $[r, s] \subset \mathbb{R}_+$ for all $a \in A_1$, then $\mathcal{C}_I \leq \frac{2s}{r+s} \cdot \mathcal{C}_C$.*

Finally, observe that unfortunately, one cannot expect to obtain a general bound for any class of distributions, and this is particularly bad in situations where most providers are "good". For some decreasing distributions, the bound becomes arbitrarily bad. Indeed, consider the case where the planner must send one unit of flow from an origin to a destination, in a two link network. One of the links is private information with cost distribution proportional to $f(c) = 1/(c + \varepsilon)$ in $[0, 1]$, while the other is public and its cost equals 1 (so $\Gamma = \{(x, y) \geq 0 : \; x + y = 1\}$). A simple calculation shows that

$$\mathcal{C}_I > 1/2 \quad \text{and} \quad \mathcal{C}_C = (\ln(1 + 1/\varepsilon))^{-1} - \varepsilon.$$

Thus, the ratio can be made arbitrarily large for small enough ε.

Furthermore, even for symmetric distributions the ratio can be arbitrarily large. To see this, consider a single good procurement auction with n sellers, i.e., $\Gamma = \{(x_1, \ldots, x_n) \geq 0 : \sum_{i=1}^n x_i = 1\}$, where each seller has a symmetric distribution putting half of the mass at or close to 0, and half at or close to 1. In this situation $\mathcal{C}_C$ is approximately $(1/2)^n$, while $\mathcal{C}_I = \mathcal{C}_{VCG}$ is roughly $(n + 1)/2^n$. The ratio grows to infinity with n.

4 Computation and Implementation

In general, implementing TVCG is no harder than solving $|A_1|$ times the original problem $\min\{c^T x : \; x \in \Gamma\}$, with the additional constraint that $x_a = 0$. In some situations this latter problem can be solved even more efficiently [6]. The situation is different for optimal mechanisms. In fact, to implement an optimal mechanism the planner must compute the assignment and the payments only for a specific cost realization. Note that this is simpler than computing the whole assignment and payment *rules*, which require the assignment and payments for every cost realization.

Given a cost realization c, Proposition 1, states that the assignment can be computed as $\min\{{c'}^T x : x \in \Gamma\}$ for some virtual nonnegative cost vector c'. This problem is the same as solving one instance of the complete information problem. However, to compute the payments $\bar{t}_a(c) = c_a \bar{x}_a(c) + \int_{c_a}^{\bar{c}_a} \bar{x}_a(t, c_{-a}) dt$ for a specific cost realization, in principle one needs to compute $\bar{x}_a(t, c_{-a})$ for all $t \in [c_a, \bar{c}_a]$. That is we need to solve $|A_1|$ parametric optimization problems of the form:

$$g_i(\theta) = \left(\arg\min_{x \in \Gamma} (c + \theta e_i)^T x \right)_i, \tag{8}$$

where $(\cdot)_i$ denotes the $i-$th component. The computational complexity of such a problem heavily depends on the structure of Γ and determines the complexity of computing the optimal mechanism under incomplete information. We now analyze three cases.

4.1 Case I: Parametric Optimization Is Easy

If the parametric optimization problem (8) can be solved in polynomial time, then the whole mechanism can be computed in polynomial time as well. This includes the case in which Γ is the set of all paths from a given source to a given sink, proved to be computationally easy in [4,6].

Observe that a wider class of problem where parametric optimization turns out to be efficient is when $\Gamma = P \subseteq [0,1]^{|A|}$, with P being an integral polytope. Of course a special case of this is $\Gamma = \{x : Ax = \mathbb{1}, x \geq 0\}$, with A totally unimodular. Shortest $s-t$ path is included in this class since it can be formulated imposing that the total flow across every $s-t$ cut equals one. Other problems in this class include minimum spanning tree and minimum perfect matching.

Proposition 5. *If $\Gamma = P \subseteq [0,1]^{|A|}$, with P an integral polytope, then (8) can be computed in polynomial time, by solving exactly two linear programming problems over P.*

Proof. From Lemma 1 $g_i(\cdot)$ is non-increasing. Also, since Γ is a $\{0,1\}$ polytope we conclude that $g_i(\theta)$ is either 0 or 1, for all $\theta \geq 0$. Therefore, to fully determine $g_i(\theta)$ (and thus obtain the optimal mechanism), it suffices to compute $\theta^* = \max\{\theta : g_i(\theta) = 1\}$.

To this end we first compute

$$Z = \min\{(c + \theta e_i)^T x : x \in \Gamma, x_i = 0\} = \min\{c_{-i}^T x_{-i} : (0, x_{-i}) \in \Gamma\}.$$

Analogously we compute

$$\theta + Z' = \min\{(c + \theta e_i)^T x : x \in \Gamma, x_i = 1\} = \theta + \min\{c_{-i}^T x_{-i} : (1, x_{-i}) \in \Gamma\}.$$

Obtaining that $\theta^* = Z' - Z$. □

Remark that the previous proposition can easily be extended to the case in which Γ is an integral polytope in $[0,K]^{|A|}$ for fixed K, or even for K of polynomial size in the input. in this case we would need to solve K linear programming problems over Γ. Furthermore, even for general Γ but satisfying that the optimal solutions to $\min_{x\in\Gamma} c^T x$ lie in $\{0,\ldots,K\}^{|A|}$, the optimal mechanism can be obtained by solving $(K+1)|A_1|$ such problems.

4.2 Case II: Optimization Is Easy but Parametric Optimization Is Hard

Even if optimizing over Γ is easy, the parametric optimization counterpart does not need to be so. For instance, for parametric linear programming, i.e., $\Gamma = \{Ax = b, x \geq 0\}$, the function $g_i(\theta)$ can attain exponentially (in $|A|$) many different values [8]. Additionally, even for more structured problems such as minimum cost flow, $g_i(\theta)$ can have a superpolynomial number of values [3].

However, since resource owners are risk-neutral, we can easily obtain a randomized mechanism that is truthful and gives in expectation the same value, therefore it is also optimal.

Indeed, for a given a cost realization c, the assignment $\bar{x}(c)$ is computed exactly as before (i.e., by solving $\min\{c'^T x : x \in \Gamma\}$), but the payments are computed using randomization. The payments to the owner of resource a, is given by $\bar{t}_a(c) = c_a \bar{x}_a(c) + (\bar{c}_a - c_a)\bar{x}_a(Y, c_{-a})$, where Y is a random variable uniformly distributed in $[c_a, \bar{c}_a]$. In expectation, which is all that matters to a risk-neutral resource owner, the latter payment equals

$$c_a \bar{x}_a(c) + (\bar{c}_a - c_a) \int_{c_a}^{\bar{c}_a} \bar{x}_a(t, c_{-a}) \cdot \frac{1}{\bar{c}_a - c_a} dt = c_a \bar{x}_a(c) + \int_{c_a}^{\bar{c}_a} \bar{x}_a(t, c_{-a}) dt,$$

and thus the mechanism is truthful and optimal. We conclude the following result.

Lemma 3. *An optimal and truthful mechanism can be implemented by solving, for each $a \in A_1$, two problems of the form* $\min\{c^T x : x \in \Gamma\}$.

Naturally the mechanism just described can be implemented in polynomial time so long as the optimization problem over Γ can be solved in polynomial time. This enables us to implement a desirable mechanism even if the parametric optimization 8 is hard. However, this mechanism introduces high risk for the resource owners. To avoid this issue we could simply take a larger number N of uniform samples Y_i and compute $\bar{t}_a(c) = c_a \bar{x}_a(c) + (\bar{c}_a - c_a) \sum_{i=1}^{N} \bar{x}_a(Y_i, c_{-a})/N$. With this the dispersion of payments will be reduced, though the computational effort will increase with N, leading to a tradeoff between risk and computational efficiency.

4.3 Case III: Optimization is Hard

We now study what happens when optimizing over Γ is NP-hard, which is the case for a large number of combinatorial problems [11]. As one may expect, computing an optimal mechanism in such a case is also hard, so we can turn to search for truthful mechanism that are approximately optimal.

Suppose that we have an algorithm ALG for solving $\min_{x \in \Gamma} c^T x$, with an approximation guarantee of β. That is an algorithm returning a solution whose cost is at most β times the optimal cost. Suppose furthermore that ALG is monotone, that is the returned solution $x_a^{\text{ALG}}(c_a, c_{-a})$ is decreasing in c_a. Then the mechanism that for each cost realization c assigns according to $x^{\text{ALG}}(c)$ is truthful. Moreover the expected cost for the planner of this mechanism is at most $\beta \cdot \mathcal{C}_I$.

References

1. Aggarwal, G., Goel, G., Mehta, A.: Efficiency of (revenue-)optimal mechanisms. In: Proceedings of the 10th ACM Conference on Electronic Commerce (EC), pp. 235–242 (2009)
2. Bulow, J., Klemperer, P.: Auctions Versus Negotiations. American Economic Review 86, 180–194 (1996)
3. Carstensen, P.: Complexity of some parametric integer and network programming problems. Mathematical Programming 26(1), 64–75 (1983)
4. Elkind, E., Sahai, A., Steiglitz, K.: Frugality in path auctions. In: Proceeding of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 701–709 (2004)

5. Hartline, J., Roughgarden, T.: Optimal Mechanism Design and Money Burning. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 75–84 (2008)
6. Hershberger, J., Suri, S.: Vickrey prices and shortest paths: What is an edge worth? In: Proceeding of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), pp. 252–259 (2001)
7. Kolmogorov, A.N., Fomin, S.V.: Introductory Real Analysis. Dover Publications, New York (1975)
8. Murty, K.: Computational complexity of parametric linear programming. Mathematical Programming 19, 213–219 (1980)
9. Myerson, R.: Optimal auction design. Mathematics of Operations Research 6(1), 58–73 (1981)
10. Nissan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge University Press, New York (2007)
11. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer, Berlin (2003)

Sequential Pivotal Mechanisms for Public Project Problems

Krzysztof R. Apt[1,2] and Arantza Estévez-Fernández[3]

[1] Centrum Wiskunde & Informatica (CWI), Amsterdam
[2] University of Amsterdam
[3] Dept. of Econometrics and Operations Research, VU University Amsterdam

Abstract. It is well-known that for several natural decision problems no budget balanced Groves mechanisms exist. This has motivated recent research on designing variants of feasible Groves mechanisms (termed as 'redistribution of VCG (Vickrey-Clarke-Groves) payments') that generate reduced deficit. With this in mind, we study sequential mechanisms and consider optimal strategies that could reduce the deficit resulting under the simultaneous mechanism. We show that such strategies exist for the sequential pivotal mechanism of the well-known public project problem. We also exhibit an optimal strategy with the property that a maximal social welfare is generated when each player follows it. Finally, we show that these strategies can be achieved by an implementation in Nash equilibrium. All proofs can be found in the full version posted in Computing Research Repository (CoRR), http://arxiv.org/abs/0810.1383

1 Introduction

1.1 Motivation

Mechanism design is concerned with designing non-cooperative games in which the participating rational players achieve the desired social outcome by reporting their types. Among the most commonly studied mechanisms are the ones in the Groves family that are based on transfer payments (taxes). For the case of efficient decision functions they are *incentive compatible*, i.e., they achieve truth-telling in dominant strategies. The special case called *pivotal mechanism* (sometimes also called *VCG* (Vickrey-Clarke-Groves) mechanism) is additionally *pay only* (i.e., each player needs to pay a tax) and hence *feasible* (i.e., the generated deficit is negative or zero).

It is well-known that for several problems incentive compatible mechanisms cannot achieve *budget balance* (which states that the generated deficit is zero), see, e.g., Chapter 23 of [14]. This has motivated recent research in designing appropriate instances of Groves mechanisms that generate a reduced deficit (or equivalently higher social welfare). These modifications are termed as 'redistribution of VCG payments'. In fact, they are variants of feasible Groves mechanisms. Notably, [3] and [9] showed that a deficit reduction is possible for the case of Vickrey auctions concerned with multiple units of a single good. On the

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 85–96, 2009.

other hand, [1] recently showed that no such deficit reduction is possible in the well-known case of the pivotal mechanism for the public project problem.

This research direction motivates our study of sequential Groves mechanisms, in particular sequential pivotal mechanism, in which players move sequentially. We face then a new situation since each player knows the types reported by the previous players. Sequential Groves mechanisms apply to a realistic situation in which there is no central authority that computes and imposes taxes and where the players move in a randomly chosen order.

1.2 Contributions

We show here that natural strategies exist in the sequential pivotal mechanism for the public project problem that generate larger social welfare than truth-telling. We also exhibit a strategy such that the social welfare is maximized when each player follows it. Finally, we show that the resulting sequential mechanisms yield an implementation in Nash equilibrium. Moreover, the vector of the latter strategies is also Pareto optimal.

To properly describe the nature of the introduced strategies we consider two concepts. An ***optimal*** strategy guarantees a player the maximum utility under the assumption that he moves simultaneously with the players who follow him. It also guarantess the player at least the same utility as truth-telling, under the assumption that the other players are truth-telling. In turn, a ***socially optimal*** strategy yields the maximal social welfare among all optimal strategies.

These concepts allow us to analyze altruistic behaviour of the players in the framework of sequential pivotal mechanism. By altruistic behaviour we mean that the players do not only care about their own utility, but also about the utility of the others.

1.3 Related Work

Ever since the seminal paper of [5] mechanism design for public goods has received a huge attention in the literature. We mention here only some representative papers the results of which provide an appropriate background for our work.

Both the continuous and discrete case of public goods have been studied. The former situation has been in particular considered in [8], where a taxation scheme has been proposed which leads to a Pareto optimal solution that can be realized in a Nash equilibrium. Sequential mechanism design for public good problems has been considered in [6], where a "Stackelberg" mechanism was proposed that combines optimal Bayes strategies with dominant strategies.

Here we study the discrete case. The situation when the decision is binary (whether to realize a public project or not) has been studied in [11], where balanced but not incentive compatible sequential mechanisms have been proposed. These mechanisms can be realized in an undominated Nash equilibrium and in subgame perfect equilibrium. Many aspects of incentive compatible mechanisms for public goods have been surveyed in [4].

The consequences of sequentiality have also been studied in the context of private contributions to public goods and in voting theory. In particular, [16] has studied the behavior of players depending on the position in which they have to take a decision and [7] has explored the relationship between simultaneous and sequential voting games. More recently, [12] has studied the problem of determining the winner in elections in which the voting takes place sequentially.

Our focus on maximizing social welfare is related to research on altruistic behaviour of the player. This subject has been studied in a number of papers in game theory, most recently in [13], where several references to earlier literature on this subject can be found. Finally, in a recent work, [2], we carried out an analogous analysis for two feasible Groves mechanisms used for single item auctions: the Vickrey auction and the Bailey-Cavallo mechanism.

1.4 Plan of the Paper

The paper is organized as follows. In the next section we recall Groves mechanisms and the pivotal mechanism by focusing on decision problems. Then, in Section 3, we introduce sequential decision problems, in particular sequential Groves mechanisms.

In the remaining sections we study the sequential pivotal mechanism for the public project problem. In Section 4 we exhibit an optimal strategy that in a limited sense simultaneously maximizes players' final utilities and another optimal strategy that maximizes the social welfare among all vectors of optimal strategies. Finally, in Section 5, we clarify the status of the optimal strategies introduced in Section 4 by showing that their vector is a Nash equilibrium w.r.t. appropriately defined preference relations on the strategy vectors, and by providing a corresponding revelation-type result. We conclude by mentioning some open problems in Section 6.

2 Preliminaries

We briefly recall the family of Groves mechanisms here. In this section we follow [10]. Let D be a set of ***decisions***, $\{1, \ldots, n\}$ be the set of players with $n \geq 2$, and for each player i let Θ_i be a set of his ***types*** and $v_i : D \times \Theta_i \rightarrow \mathbb{R}$ be his (***initial***) ***utility function***.

A ***decision rule*** is a function $f : \Theta \rightarrow D$, where $\Theta := \Theta_1 \times \cdots \times \Theta_n$. It is called ***efficient*** if for all $\theta \in \Theta$ and $d' \in D$

$$\sum_{i=1}^{n} v_i(f(\theta), \theta_i) \geq \sum_{i=1}^{n} v_i(d', \theta_i).$$

We call the tuple $(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$ a ***decision problem***.

Recall that a ***direct mechanism*** is obtained by transforming the initial decision problem $(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$ as follows:

– the set of decisions is $D \times \mathbb{R}^n$,

- the decision rule is a function $(f,t) : \Theta \rightarrow D \times \mathbb{R}^n$, where $t : \Theta \rightarrow \mathbb{R}^n$ and $(f,t)(\theta) := (f(\theta), t(\theta))$,
- each ***final utility function*** for player i is a function $u_i : D \times \mathbb{R}^n \times \Theta_i \rightarrow \mathbb{R}$ defined by $u_i(d, t_1, \ldots, t_n, \theta_i) := v_i(d, \theta_i) + t_i$.

We call then $\sum_{i=1}^{n} u_i((f,t)(\theta), \theta_i)$ the corresponding ***social welfare*** and refer to t as the ***tax function***.

A direct mechanism with tax function t is called

- (***dominant strategy***) ***incentive compatible*** if for all $\theta \in \Theta$, $i \in \{1, \ldots, n\}$ and $\theta_i' \in \Theta_i$
$$u_i((f,t)(\theta_i, \theta_{-i}), \theta_i) \geq u_i((f,t)(\theta_i', \theta_{-i}), \theta_i),$$
- ***budget balanced*** if $\sum_{i=1}^{n} t_i(\theta) = 0$ for all θ,
- ***feasible*** if $\sum_{i=1}^{n} t_i(\theta) \leq 0$ for all θ,
- ***pay only*** if $t_i(\theta) \leq 0$ for all θ and all $i \in \{1, \ldots, n\}$.

Each ***Groves mechanism*** is obtained by using the tax function $t := (t_1, \ldots, t_n)$, where[1] for all $i \in \{1, \ldots, n\}$

$$t_i(\theta) := \sum_{j \neq i} v_j(f(\theta), \theta_j) + h_i(\theta_{-i}),$$

with $h_i : \Theta_{-i} \rightarrow \mathbb{R}$ an arbitrary function.

Finally, we recall the following crucial result.

Groves Theorem. Consider a decision problem with an efficient decision rule f. Then each Groves mechanism is incentive compatible.

A special case of Groves mechanism is the ***pivotal mechanism***, which is a pay only mechanism obtained by $h_i(\theta_{-i}) := -\max_{d \in D} \sum_{j \neq i} v_j(d, \theta_j)$.

Direct mechanisms for a given decision problem can be compared w.r.t. the social welfare they entail. More precisely, given a decision problem

$$(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$$

and direct mechanisms (determined by the sequences of tax functions) t and t' we say that t' ***welfare dominates*** t if

- for all $\theta \in \Theta$
$$\sum_{i=1}^{n} u_i((f,t)(\theta), \theta_i) \leq \sum_{i=1}^{n} u_i((f,t')(\theta), \theta_i),$$
- for some $\theta \in \Theta$
$$\sum_{i=1}^{n} u_i((f,t)(\theta), \theta_i) < \sum_{i=1}^{n} u_i((f,t')(\theta), \theta_i).$$

In this paper we analyze the following well-known decision problem, originally due to [5], and extensively discussed in the economic literature, see, e.g. [14,10].

[1] Here and below $\sum_{j \neq i}$ is a shorthand for the summation over all $j \in \{1, \ldots, n\}$, $j \neq i$.

Public project problem
Consider $(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$, where

- $D = \{0, 1\}$ (reflecting whether a project is cancelled or takes place),
- for all $i \in \{1, \ldots, n\}$, $\Theta_i = [0, c]$, where $c > 0$,
- for all $i \in \{1, \ldots, n\}$, $v_i(d, \theta_i) := d(\theta_i - \frac{c}{n})$,
- $f(\theta) := \begin{cases} 1 & \text{if } \sum_{i=1}^{n} \theta_i \geq c \\ 0 & \text{otherwise} \end{cases}$

In this setting c is the cost of the project, $\frac{c}{n}$ is the cost share of the project for each player, and θ_i is the value of the project for player i. Note that the decision rule f is efficient since $\sum_{i=1}^{n} v_i(d, \theta_i) = d(\sum_{i=1}^{n} \theta_i - c)$.

It is well-known that for no $n \geq 2$ and $c > 0$ an incentive compatible direct mechanism for the public project problem exists that is budget balanced, see, e.g. [14, page 861-862]. It is then natural to search for incentive compatible direct mechanisms that generate a smaller deficit than the one obtained by the pivotal mechanism. However, the following optimality result concerning the pivotal mechanism, recently established in [1], dashed hope.

Theorem 1. *In the public project problem there exists no feasible incentive compatible direct mechanism that welfare dominates the pivotal mechanism.* □

Our aim is to show that when the original setting of the public project problem is changed to one where all players announce their types sequentially in a random order, then the deficit can be reduced.

3 Sequential Decision Problems

In this section we introduce sequential decision problems. For notational simplicity, and without loss of generality, we assume that players sequentially report their types in the order $1, \ldots, n$. To capture this type of situations, given a decision problem $\mathcal{D} := (D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$, we assume that successively stages $1, \ldots, n$ take place, where in stage i player i announces a type θ'_i to the *other players*. After stage n this yields a joint type $\theta' := (\theta'_1, \ldots, \theta'_n)$. Then each player takes the decision $d := f(\theta')$.

We call the resulting situation a ***sequential decision problem*** or more specifically, a ***sequential version of*** $\mathcal{D}$. Note that in a sequential decision problem a central planner may not exist and decisions may be taken by the players themselves. Each player i *knows* the types announced by players $1, \ldots, i-1$, so that he can use this information to decide which type to announce. To properly describe this situation we need to specify what is a strategy in this setting. A ***strategy*** of player i in the sequential version of $\mathcal{D}$ is a function

$$s_i : \Theta_1 \times \ldots \times \Theta_i \rightarrow \Theta_i.$$

In this context ***truth-telling***, as a strategy, is represented by the projection function $\pi_i(\cdot)$, defined by $\pi_i(\theta_1, \ldots, \theta_i) := \theta_i$.

From now on, we consider a direct mechanism

$$\mathcal{D} := (D \times \mathbb{R}^n, \Theta_1, \ldots, \Theta_n, u_1, \ldots, u_n, (f, t))$$

and mainly focus on Groves mechanisms.

We assume that in the considered sequential decision problem each player uses a strategy $s_i(\cdot)$ to select the type he will announce. We say that strategy $s_i(\cdot)$ of player i is ***optimal*** in the sequential version of $\mathcal{D}$ if for all $\theta \in \Theta$ and $\theta'_i \in \Theta_i$

$$u_i((f,t)(s_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_i) \geq u_i((f,t)(\theta'_i, \theta_{-i}), \theta_i).$$

Call a strategy of player j ***memoryless*** if it does not depend on the types of players $1, \ldots, j-1$. Then a strategy $s_i(\cdot)$ of player i is optimal if for all $\theta \in \Theta$ it yields a best response to all joint strategies of players $j \neq i$ under the assumption that players $i+1, \ldots, n$ use memoryless strategies or move jointly with player i. In particular, an optimal strategy is a best response to the truth-telling by players $j \neq i$.

A particular case of sequential decision problems are sequential Groves mechanisms. The following direct consequence of Groves Theorem provides us with a simple method of determining whether a strategy is optimal in such a mechanism.

Lemma 1. *Let $(D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$ be a decision problem with efficient decision rule f. Suppose that $s_i(\cdot)$ is a strategy for player i such that for all $\theta \in \Theta$, $f(s_i(\theta_1, \ldots, \theta_i), \theta_{-i}) = f(\theta_i, \theta_{-i})$. Then $s_i(\cdot)$ is optimal in each sequential Groves mechanism $(D \times \mathbb{R}^n, \Theta_1, \ldots, \Theta_n, u_1, \ldots, u_n, (f, t))$.* □

In particular, when the decision rule is efficient, the truth-telling strategy $\pi_i(\cdot)$ is optimal in each sequential Groves mechanism.

We are interested in maximizing the social welfare. This motivates the following notion. We say that strategy $s_i(\cdot)$ of player i is ***socially optimal*** in the sequential version of $\mathcal{D}$ if it is optimal and for all optimal strategies $s'_i(\cdot)$ of player i and all $\theta \in \Theta$ we have

$$\textstyle\sum_{j=1}^n u_j((f,t)(s_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_j) \geq \sum_{j=1}^n u_j((f,t)(s'_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_j).$$

Hence a socially optimal strategy of player i yields the maximal social welfare among all optimal strategies, under the assumption that players $i+1, \ldots, n$ use memoryless strategies or move jointly.

Consider now a sequential version of a given direct mechanism

$$(D \times \mathbb{R}^n, \Theta_1, \ldots, \Theta_n, u_1, \ldots, u_n, (f, t))$$

and assume that each player i receives a type $\theta_i \in \Theta_i$ and follows a strategy $s_i(\cdot)$. The resulting social welfare is then

$$SW(\theta, s(\cdot)) := \sum_{j=1}^n u_j((f,t)([s(\cdot), \theta]), \theta_j),$$

where $s(\cdot) := (s_1(\cdot), \ldots, s_n(\cdot))$ and $[s(\cdot), \theta]$ is defined inductively by $[s(\cdot), \theta]_1 := s_1(\theta_1)$ and $[s(\cdot), \theta]_{i+1} := s_{i+1}([s(\cdot), \theta]_1, \ldots, [s(\cdot), \theta]_i, \theta_{i+1})$.

In general, if player i assumes that he moves jointly with players $i+1, \ldots, n$ he will choose an optimal strategy. And if additionally he wants to maximize the social welfare, he will choose a socially optimal strategy (if it exists). In the next section we shall see that for the public project problem a sequence of socially optimal strategies can be found for which the resulting social welfare is always maximal. In general, we only have the following limited result.

Lemma 2. *Consider a direct mechanism* $(D \times \mathbb{R}^n, \Theta_1, \ldots, \Theta_n, u_1, \ldots, u_n, (f,t))$ *and let* $s_n(\cdot)$ *be a socially optimal strategy for player* n*. Then*

$$SW(\theta, (s'_{-n}(\cdot), s_n(\cdot))) \geq SW(\theta, s'(\cdot))$$

for all $\theta \in \Theta$ *and vectors* $s'(\cdot)$ *of optimal players' strategies.*

Proof. Directly by the definition of a socially optimal strategy. □

4 Public Project Problem

In what follows, we focus on the special case of sequential pivotal mechanisms for the public project problem. First, the following theorem gives an optimal strategy for player i that may differ from truth-telling. Part (ii) shows that, under certain natural conditions, this strategy simultaneously maximizes the final utility of every other player.

Theorem 2. *Let* $\mathcal{D}$ *be a public project problem. Let*

$$s_i(\theta_1, \ldots, \theta_i) := \begin{cases} \theta_i & \text{if } \sum_{j=1}^{i} \theta_j < c \text{ and } i < n, \\ 0 & \text{if } \sum_{j=1}^{i} \theta_j < c \text{ and } i = n, \\ c & \text{if } \sum_{j=1}^{i} \theta_j \geq c \end{cases}$$

be strategy for player i*. Then*

(i) $s_i(\cdot)$ *is optimal for player* i *in the sequential pivotal mechanism,*
(ii) for all $\theta \in \Theta$ *and* $\theta'_i \in \Theta_i$ *such that* $s_i(\theta_1, \ldots, \theta_i) \neq \theta_i$ *and* $f(\theta'_i, \theta_{-i}) = f(\theta_i, \theta_{-i})$ *we have for all* $j \neq i$

$$u_j((f,t)(s_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_j) \geq u_j((f,t)(\theta'_i, \theta_{-i}), \theta_j).$$

In part (ii) θ_{-i} are the types submitted by players $j \neq i$ and θ_i is the type received by player i. So part (ii) states that if strategy $s_i(\cdot)$ of player i deviates from truth-telling ($s_i(\theta_1, \ldots, \theta_i) \neq \theta_i$) and the players who follow i use memoryless strategies (so in particular, the types they submit do not depend on the type submitted by player i), then player i simultaneously maximizes the final utility of the other players (and hence the social welfare). This happens under the assumption that player i submits a type that does not alter the decision to be taken.

Table 1. Pivotal mechanism

player	type	submitted type	tax	u_i
A	110	110	-10	0
B	80	80	0	-20
C	110	110	-10	0

Table 2. Sequential pivotal mechanism

player	type	submitted type	tax	u_i
A	110	110	0	10
B	80	80	0	-20
C	110	300	-10	0

When each player follows strategy $s_i(\cdot)$, always the same decision is taken as when each player is truthful, independently on the players' order. Additionally, by part (ii) of Theorem 2 with $\theta'_i = \theta_i$, social welfare weakly increases. The following example shows that sometimes a strictly larger social welfare can be achieved.

Example 1. Assume that $c = 300$ and that there are three players, A, B and C. Table 1 illustrates the situation in the case of pivotal mechanism. In Table 2 we assume that the players submit their types in the order A, B, C. Here the social welfare increases from -20 to -10. □

However, as Table 2 shows, budget balance does not need to be achieved. The following result shows that an order can always be found that yields budget balancedness.

Theorem 3. *Let $\mathcal{D}$ be a public project problem with the sequential pivotal mechanism. For all $c > 0, n \geq 2$ and $\theta \in \Theta$ there exists a permutation of players such that when each player i follows strategy $s_i(\cdot)$ of Theorem 2, budget balance is achieved.*

Proof. (Sketch). Recall that in the pivotal mechanism, given the sequence of types θ, a player i is called ***pivotal*** if $t_i(\theta) \neq 0$. First we show that not all players can be pivotal. Then we show that the desired permutation is the one in which the last player is not pivotal. □

For instance, in Example 1 when the order is A, C, B or C, A, B, the decision is taken with no taxes incurred, i.e., budget balance is then achieved.

In Theorem 2(ii) we seem to be maximizing the social welfare. However, this is not the case because we assume there that each player submits a type that does not alter the decision to be taken. In fact, strategy $s_i(\cdot)$ of Theorem 2 is not socially optimal.

The following theorem provides a socially optimal strategy that in some circumstances yields a higher social welfare than the above strategy.

Theorem 4. *Let $\mathcal{D}$ be a public project problem. Let*

$$s_i(\theta_1, \ldots, \theta_i) := \begin{cases} \theta_i & \text{if } \sum_{j=1}^{i} \theta_j < c \text{ and } i < n, \\ 0 & \text{if } \sum_{j=1}^{i} \theta_j < c \text{ and } i = n, \\ 0 & \text{if } \sum_{j=1}^{i} \theta_j = c,\ \theta_i > \frac{c}{n} \text{ and } i = n, \\ c & \text{otherwise} \end{cases}$$

be a strategy for player i. Then

(i) $s_i(\cdot)$ is socially optimal for player i in the sequential pivotal mechanism,
(ii) for all $\theta \in \Theta$ and vectors $s'(\cdot)$ of optimal players' strategies,

$$SW(\theta, s(\cdot)) \geq SW(\theta, s'(\cdot)),$$

where $s(\cdot)$ is the vector of strategies $s_i(\cdot)$.

The remarkable thing about the above strategy $s_i(\cdot)$ is that when $\sum_{j=1}^{n} \theta_j = c$ and $\theta_n > \frac{c}{n}$, player n submits type 0, as a result of which the project does not take place. To illustrate this situation reconsider Example 1. When the players submit their types sequentially in order A, B, C following the above strategy $s_i(\cdot)$, then player C submits 0. The resulting social welfare is 0 as opposed to -10 which results when all players follow strategy $s_i(\cdot)$ of Theorem 2 (see Table 2). This also shows that the latter strategy is not socially optimal.

However, in general strategy $s_i(\cdot)$ of Theorem 4 does not need to ensure budget balance.

Example 2. Suppose that there are three players, A, B, and C, whose true types are 60, 70, and 250, respectively, while c remains 300. When the players submit their types following strategy $s_i(\cdot)$ of Theorem 4, we get the situation summarized in Table 3.

Table 3. Sequential pivotal mechanism

player	type	submitted type	tax	u_i
A	60	60	0	-40
B	70	70	0	-30
C	250	300	-70	80

Here the same decision is taken as when each player is truthful and in both situations the deficit is -70. □

On other other hand, part (ii) shows that when we limit ourselves to optimal strategies and each player follows the introduced strategy $s_i(\cdot)$, then a maximal social welfare results. The restriction to the vectors of optimal strategies is necessary. Indeed, Table 3 of Example 2 shows that when the order is A, B, C and each player follows the strategy $s_i(\cdot)$ of Theorem 4, then the resulting social welfare is $380 - 300 - 70 = 10$. However, when player B submits 300, then player C pays no tax and the resulting social welfare is higher, namely $380 - 300 = 80$.

5 Comments on a Nash Implementation

The sequential mechanisms here considered circumvent the limitations of the customary, simultaneous, Groves mechanisms. This and the fact that we maximize social welfare by using strategies that deviate from truth-telling requires some clarification. First of all, we can explain these sequential mechanisms by turning them into simultaneous ones as follows.

We assume that each player i receives a type $\theta_i \in \Theta_i$ and subsequently submits a function $r_i : \Theta_1 \times \ldots \times \Theta_{i-1} \rightarrow \Theta_i$ instead of a type $\theta'_i \in \Theta_i$. (In particular, player 1 submits a type, i.e., $r_1(\cdot) \in \Theta_1$.) The submissions are simultaneous. Then the behaviour of player i can be described by a strategy $s_i : \Theta_1 \times \ldots \times \Theta_i \rightarrow \Theta_i$ which when applied to the received type θ_i yields the function $s_i(\cdot, \theta_i) : \Theta_1 \times \ldots \times \Theta_{i-1} \rightarrow \Theta_i$ that player i submits. Then θ and the vector $s(\cdot) := (s_1(\cdot), \ldots, s_n(\cdot))$ of strategies that the players follow yield an element $[s(\cdot), \theta]$ of Θ, where, recall, $[s(\cdot), \theta]_1 := s_1(\theta_1)$ and $[s(\cdot), \theta]_{i+1} := s_{i+1}([s(\cdot), \theta]_1, \ldots, [s(\cdot), \theta]_i, \theta_{i+1})$.

Given a decision problem $\mathcal{D} := (D, \Theta_1, \ldots, \Theta_n, v_1, \ldots, v_n, f)$ and two strategies $s_i(\cdot)$ and $s'_i(\cdot)$ of player i in the sequential version of $\mathcal{D}$, we write

$$\begin{aligned} s_i(\cdot) \geq_d s'_i(\cdot) \text{ iff for all } \theta \in \Theta \\ v_i(f(s_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_i) \geq v_i(f(s'_i(\theta_1, \ldots, \theta_i), \theta_{-i}), \theta_i). \end{aligned}$$

We write $s_i(\cdot) >_d s'_i(\cdot)$ if additionally one of these inequalities is strict, and we write $s_i(\cdot) =_d s'_i(\cdot)$ if all these inequalities are equalities.

Note that $s_i(\cdot) \geq_d s'_i(\cdot)$ for all strategies $s'_i(\cdot)$ of player i iff strategy $s_i(\cdot)$ of player i is optimal in the sequential version of $\mathcal{D}$.

Next, we define for all $i \in \{1, \ldots, n\}$ a preference relation $\succeq_i$ on the vectors of players' strategies by writing

$$\begin{aligned} s(\cdot) \succeq_i s'(\cdot) \text{ iff } & s_i(\cdot) >_d s'_i(\cdot) \text{ or} \\ & (s_i(\cdot) =_d s'_i(\cdot) \text{ and} \\ & \text{for all } \theta \in \Theta,\ v_i(f([s(\cdot), \theta]), \theta_i) \geq v_i(f([s'(\cdot), \theta]), \theta_i)). \end{aligned}$$

We now say that a joint strategy $s(\cdot)$ is a ***Nash equilibrium*** in the sequential version of $\mathcal{D}$ if for all $i \in \{1, \ldots, n\}$ and all strategies $s'_i(\cdot)$ of player i we have

$$(s_i(\cdot), s_{-i}(\cdot)) \succeq_i (s'_i(\cdot), s_{-i}(\cdot)).$$

The following result clarifies the status of the strategies introduced in Theorems 2 and 4.

Theorem 5. *Let $\mathcal{D}$ be a public project problem.*

(i) Each of the vectors $s(\cdot)$ of strategies defined in Theorems 2 and 4, respectively, is a Nash equilibrium in the corresponding sequential version of the pivotal mechanism.

(ii) The vector $s(\cdot)$ of Theorem 2 is Pareto optimal in the universe of optimal strategies, in the sense that for all $\theta \in \Theta$ the resulting social welfare $SW(\theta, s(\cdot))$ is maximal among all vectors of optimal players' strategies.

This result shows that the improvement in terms of the maximization of the social welfare over the Groves mechanism is achieved by weakening the implementation in dominant strategies (see Groves Theorem) to an implementation in Nash equilibrium (in the universe of optimal strategies).

The above definition of the $\succeq_i$ relation uses the $>_d$ relation to ensure that in the definition of Nash equilibrium the deviations to non-optimal strategies are trivially discarded. This ruling out of non-optimal strategies is necessary. Indeed, when $\theta_i > \frac{c}{n}$, with $i < n$, and $\sum_{j=1}^{n} \theta_j < c$, then player's i final utility increases from 0 to $\theta_i - \frac{c}{n}$ when he deviates from any of the two strategies considered in Theorem 5 to the strategy

$$s_i(\theta_1, \ldots, \theta_i) := \begin{cases} 0 & \text{if } \theta_i \leq \frac{c}{n} \\ c & \text{otherwise.} \end{cases}$$

Recall now that the well-known revelation principle (see, e.g., [15]) states that every mechanism can be realized as a (simultaneous) direct mechanism in which truth-telling is the optimal strategy. We now show that using any Nash equilibrium $(s_1(\cdot), \ldots, s_n(\cdot))$ of Theorem 5 we can construct a revelation-type simultaneous mechanism in which the vector $(\pi_1(\cdot), \ldots, \pi_n(\cdot))$ of the projection functions forms a Nash equilibrium. (Recall that the $\pi_i(\cdot)$ function corresponds in the sequential setting to truth-telling by player i.) This mechanism is constructed using the following preference relations $\succeq_i^*$ on the vectors of players' strategies:

$$s'(\cdot) \succeq_i^* s''(\cdot) \text{ iff}$$
$$(s_1(\cdot) \circ s'_1(\cdot), \ldots, s_n(\cdot) \circ s'_n(\cdot)) \succeq_i (s_1(\cdot) \circ s''_1(\cdot), \ldots, s_n(\cdot) \circ s''_n(\cdot)),$$

where strategy $s_i(\cdot) \circ s'_i(\cdot)$ of player i is defined by

$$(s_i(\cdot) \circ s'_i(\cdot))(\theta_1, \ldots, \theta_i) := s_i(\theta_1, \ldots, \theta_{i-1}, s'_i(\theta_1, \ldots, \theta_i)).$$

Theorem 6. *Let $\mathcal{D}$ be a public project problem. The vector $(\pi_1(\cdot), \ldots, \pi_n(\cdot))$ of projection strategies is a Nash equilibrium in the corresponding sequential version of the pivotal mechanism, where we use the preference relations $\succeq_1^*, \ldots, \succeq_n^*$.*

Proof. Note that for all $j \in \{1, \ldots, n\}$, $s_j(\cdot) \circ \pi_j(\cdot) = s_j(\cdot)$. Then

$$(\pi_i(\cdot), \pi_{-i}(\cdot)) \succeq_i^* (s'_i(\cdot), \pi_{-i}(\cdot)) \text{ iff } (s_i(\cdot), s_{-i}(\cdot)) \succeq_i (s_i(\cdot) \circ s'_i(\cdot), s_{-i}(\cdot)),$$

so the result holds by Theorem 5(i). □

6 Concluding Remarks

As already mentioned, no budget balanced Groves mechanisms exist for the public project. We have investigated here to what extent the unavoidable deficit can be reduced when players move sequentially. By focusing on socially optimal strategies we have incorporated into our analysis altruistic behaviour of the players.

The results here established hold for the sequential pivotal mechanism. Some of them, but not all, can be generalized to sequential Groves mechanisms. More specifically, the strategies introduced in Theorems 2 and 4 are also optimal in arbitrary sequential Groves mechanisms. The reason is the following observation.

Note 1. Fix an initial decision problem and consider two Groves mechanisms (with tax functions) t and t'. A strategy of player i is optimal in the sequential version of t iff it is optimal in the sequential version of t'.

How to generalize the remaining claims of Theorems 2 and 4 to other sequential Groves mechanisms remains an interesting open problem.

References

1. Apt, K.R., Conitzer, V., Guo, M., Markakis, V.: Welfare undominated groves mechanisms. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 426–437. Springer, Heidelberg (2008)
2. Apt, K.R., Markakis, V.: Optimal strategies in sequential bidding. In: Proceedings AAMAS 2009, pp. 1189–1190. IFAAMAS (2009)
3. Cavallo, R.: Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In: Proceedings AAMAS 2006, pp. 882–889. ACM Press, New York (2006)
4. Chen, Y.: Incentive-compatible mechanisms for pure public goods: A survey of experimental research. In: Plott, C., Smith, V. (eds.) Handbook of Experimental Economics Results. Elsevier, Amsterdam (2008) (forthcoming)
5. Clarke, E.H.: Multipart pricing of public goods. Public Choice 11, 17–33 (1971)
6. Crémer, J., Riordan, M.H.: A sequential solution to the public goods problem. Econometrica 53(1), 77–84 (1985)
7. Dekel, E., Piccione, M.: Sequential voting procedures in symmetric binary elections. Journal of Political Economy 108(1), 34–55 (2000)
8. Groves, T., Ledyard, J.: Optimal allocation of public goods: A solution to the "free rider" problem. Econometrica 45(4), 783–809 (1977)
9. Guo, M., Conitzer, V.: Worst-case optimal redistribution of VCG payments. In: EC 2007: Proceedings of the 8th ACM conference on Electronic commerce, pp. 30–39. ACM Press, New York (2007)
10. Jackson, M.: Mechanism theory. In: Derigs, U. (ed.) Encyclopedia of Life Support Systems. EOLSS Publishers, Oxford (2003)
11. Jackson, M., Moulin, H.: Implementing a public project and distributing its costs. Journal of Economic Theory (57), 125–140 (1992)
12. Lang, J., Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Winner determination in sequential majority voting. In: Proceedings IJCAI 2007, pp. 1372–1377 (2007)
13. De Marco, G., Morgan, J.: Slightly altruistic equilibria in normal form games. In: Working paper No 185, Center for Studies in Economics and Finance, University of Salerno, Italy (2007), http://www.csef.it/WP/wp185.pdf
14. Mas-Collel, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, Oxford (1995)
15. Myerson, R.B.: Game Theory: Analysis of Conflict. Harvard Univ. Press, Cambridge (1991)
16. Varian, H.: Sequential provision of public goods. Journal of Public Economics 53(2), 165–186 (1994)

Characterizing the Existence of Potential Functions in Weighted Congestion Games

Tobias Harks, Max Klimm, and Rolf H. Möhring

Institut für Mathematik, Technische Universität Berlin
{harks,klimm,moehring}@math.tu-berlin.de

Abstract. Since the pioneering paper of Rosenthal a lot of work has been done in order to determine classes of games that admit a potential. First, we study the existence of potential functions for weighted congestion games. Let $\mathcal{C}$ be an arbitrary set of locally bounded functions and let $\mathcal{G}(\mathcal{C})$ be the set of weighted congestion games with cost functions in $\mathcal{C}$. We show that every weighted congestion game $G \in \mathcal{G}(\mathcal{C})$ admits an exact potential if and only if $\mathcal{C}$ contains only affine functions. We also give a similar characterization for weighted potentials with the difference that here $\mathcal{C}$ consists either of affine functions or of certain exponential functions. We finally extend our characterizations to weighted congestion games with facility-dependent demands and elastic demands, respectively.

1 Introduction

In many situations, the state of a system is determined by a large number of independent agents, each pursuing selfish goals optimizing an individual objective function. A natural framework for analyzing such decentralized systems are noncooperative games. It is well known that an equilibrium point in pure strategies (if it exists) need not optimize the social welfare as individual incentives are not always compatible with social objectives. Fundamental goals in algorithmic game theory are to decide whether a Nash equilibrium in pure strategies (PNE for short) exists, how efficient it is in the worst case, and how fast an algorithm (or protocol) converges to an equilibrium.

One of the most successful approaches in accomplishing these goals is the potential function approach initiated by Rosenthal [24] and generalized by Monderer and Shapley in [22]: one defines a function P on the set of possible strategies of the game and shows that every strictly improving move by one defecting player strictly reduces (increases) the value of P. Since the set of outcomes of such a game is finite, every sequence of improving moves reaches a PNE. In particular, the global minimum (maximum) of P is a PNE. A function P with the property above is called a *potential function* of the game. If one can associate a weight w_i to each player such that $w_i P$ decreases about the same value as the private cost of the defecting player i, then P is called a *weighted potential*. If, in addition, $w_i = 1$ for each player, then P is called an *exact potential*.

1.1 Framework

The first part of this paper studies the existence of potential functions in weighted congestion games (Definition 4). Congestion games, as introduced by Rosenthal [24],

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 97–108, 2009.

model the interaction of a finite set of strategic agents that compete over a finite set of facilities. A pure strategy of each player is a set of facilities. We consider cost minimization games. Here, the cost of facility f is given by a real-valued cost function c_f that depends on the number of players using f and the private cost of every player equals the sum of the costs of the facilities in the strategy that she chooses.[1] Rosenthal [24] proved in a seminal paper that such congestion games always admit a PNE by showing these games posses an exact potential function.

In a *weighted congestion game*, every player has a demand $d_i \in \mathbb{R}_+$ that she places on the chosen facilities. The cost of a facility is a function of the total demand of the facility. In contrast to unweighted congestion games, weighted congestion games, even with two players, do not always admit a PNE, see the examples given by Fotakis et al. [11], Goemans et al. [14], and Libman and Orda [18].

On the positive side, Fotakis et al. [11,12] proved that every weighted congestion game with affine cost functions possesses an exact potential function and thus, a PNE. Panagopoulou and Spirakis [23] proved existence of a weighted potential function for the set of exponential cost functions.

The results of [11,12] and [23] are particularly appealing as they establish existence of a potential function *independent* of the underlying game structure, that is, *independent* of the underlying strategy set, demand vector, and number of players, respectively. To further stress this independence property, we rephrase the result of Fotakis et al. as follows: Let C be a set of affine cost functions and let $\mathcal{G}(C)$ be the set of *all* weighted congestion games with cost functions in C. Then, *every* game in $\mathcal{G}(C)$ possesses an exact potential.

A natural open question is to decide whether there are further functions guaranteeing the existence of an exact or weighted potential. We thus investigate the following question: How large is the class C of (continuous) cost functions such that every game in the set of weighted congestion games $\mathcal{G}(C)$ with cost functions in C does admit a potential function and hence a PNE?

Before we outline our results we present related work and explain, why it is important to characterize weighted congestion games admitting a potential function.

1.2 Related Work

Fundamental issues in algorithmic game theory are the computability of Nash equilibria and the design of distributed dynamics (for instance best-response) that provably converge in reasonable time to a Nash equilibrium (in pure or mixed strategies).

Monderer and Shapley [22] formalized Rosenthal's approach of using potential functions to determine the existence of PNE. Furthermore, they show that one-side better response dynamics always converge to a PNE provided the game is finite and admits a potential. In addition, they proved that weighted potential games have other desirable properties, e.g., the Fictitious Play Process converges to a PNE [21]. For recent progress on convergence towards approximate Nash equilibria using potential functions, see Awerbuch et al. [4] and Fotakis et al. [10].

[1] Since we allow the cost of a facility to be positive or negative, we also cover maximization games.

Fabrikant et al. [9] proved that one can efficiently compute a PNE for symmetric network congestion games with nondecreasing cost functions. Their proof uses a potential function argument, similar to Rosenthal [24]. Fotakis et al. [11] proved that one can compute a PNE for weighted network games with affine cost (with nonnegative coefficients) in pseudo-polynomial time (again using a potential function).

Milchtaich [20] introduced weighted congestion games with player-specific cost functions. Among other results, he presented a game on 3 parallel links with 3 players, which does not possess a PNE. On the other hand, he proved that such games with 2 players do possess a PNE. Ackermann et al. [1] characterized conditions on the strategy space in weighted congestion games that guarantee the existence of PNE. They also considered the case of player-specific cost functions.

Gairing et al. [13] derive a potential function for the case of weighted congestion games with player-specific linear latency functions (without a constant term). Mavronicolas et al. [19] prove that every unweighted congestion game with player-specific (additive or multiplicative) constants on parallel links has an ordinal potential. Even-Dar et al. [8] consider a variety of load balancing games with makespan objectives and prove among other results that games on unrelated machines possess a generalized ordinal potential function. For related results, see the survey by Vöcking [25] and references therein.

Potential functions also play a central role in Shapley cost sharing games with weighted players, which are special cases of weighted congestion games, see Anshelevich et al. [3] and Albers et al. [2]. In the variant with weighted players, each player i has a demand d_i that she wishes to place on each facility of an allowable subset of facilities (e.g., a path in a network connecting her source node s_i to her terminal node t_i). When facility $f \in F$ is stressed with a load of $\ell_f(x)$ in strategy profile x, it causes a cost of $k_f(\ell_f(x))$. Under Shapley cost sharing, this cost is shared linearly with respect to the demands among the users. Thus the cost of player i for using facility f is defined as $c_{i,f}(x) = k_f(\ell_f(x))\, d_i/\ell_f(x)$ and clearly, the private cost of player i in strategy profile x is given as $\pi_i(x) = \sum_{f \in x_i} c_{i,f}(x)$. For the unweighted case ($d_i = 1, i \in N$), Anshelevich et al. [3] proved existence of PNE and derived bounds on the price of stability using a potential function argument. This argument fails in general for games with weighted players, see the counterexamples given by Chen and Roughgarden [5]. Determining subclasses of Shapley cost sharing games with weighted players that admit a potential, however, is an open problem that we address in this paper.

1.3 Our Results for Weighted Congestion Games

Our first two results provide a characterization of the existence of exact and weighted potential functions for the set of weighted congestion games with locally bounded and continuous cost functions, respectively. Let C be an arbitrary set of locally bounded functions and let $\mathcal{G}(C)$ be the set of weighted congestion games with cost functions in C. We show that every weighted congestion game $G \in \mathcal{G}(C)$ admits an exact potential if and only if C contains only affine functions. For an arbitrary set C of continuous functions, we show that every weighted congestion game $G \in \mathcal{G}(C)$ possesses a weighted potential if and only if exactly one of the following cases hold: (i) C contains only affine functions; (*ii*) C contains only exponential functions such that $c(\ell) = a_c\, e^{\phi\,\ell} + b_c$ for some $a_c, b_c, \phi \in \mathbb{R}$, where a_c and b_c may depend on c, while ϕ must be equal for every $c \in C$.

We additionally show that the above characterizations for exact and weighted potentials are valid even if we restrict the set $\mathcal{G}(C)$ to two-player games (three-player games for weighted potentials), three-facility games (four-facility games for weighted potentials), games with symmetric strategies, games with singleton strategies, games with integral demands. Moreover, we derive a result for weighted congestion games where each facility is contained in the strategy set of at most two players, showing that every such game with cost functions in C admits a weighted potential if $C = \{(c : \mathbb{R}_+ \to \mathbb{R}) : c(x) = a f(x) + b,\ a, b \in \mathbb{R}\}$, where $f : \mathbb{R}_+ \to \mathbb{R}$ is a strictly monotonic function.

Our results have a series of consequences. First, using a result of Monderer and Shapley [22, Lemma 2.10], our characterization of weighted potentials in weighted congestion games carries over to the mixed extension of weighted congestion games.

Second, we obtain the following characterizations for Shapley cost sharing games. Let $\mathcal{K}$ be a set of continuous functions. Then, the set $\mathcal{S}(\mathcal{K})$ of Shapley cost sharing games with weighted players and construction cost functions in $\mathcal{K}$ are weighted potential games if and only if $\mathcal{K}$ contains either quadratic construction cost functions ($k(\ell) = a_k\,\ell^2 + b_k\,\ell$) or functions of type $k(\ell) = a_k\, e^{\phi\,\ell}\,\ell + b_k\,\ell$ for some $a_k, b_k, \phi \in \mathbb{R}$, where a_k and b_k may depend on k, while ϕ must be equal for every $k \in \mathcal{K}$. Notice that these results hold for arbitrary coefficients $a_k, b_k, \phi \in \mathbb{R}$. Thus, we obtain the existence of PNE for a family of games with nondecreasing and strictly concave construction costs modeling the effect of economies of scale.

1.4 Our Results for Extended Models

In the second part of this paper, we introduce two non-trivial extensions of weighted congestion games.

First, we study weighted congestion games with *facility-dependent* demands, that is, the demand $d_{i,f}$ of player i depends on the facility f. These games contain, among others, scheduling games on identical, restricted, related and unrelated machines. In contrast to classical load balancing games, we do not consider makespan objectives. In our model, the private cost of a player is a function of the machine load multiplied with the demand of the player.

We show the following: Let C be a set of continuous functions and let $\mathcal{G}^{fd}(C)$ denote the set of weighted congestion games with facility-dependent demands and cost functions in C. Every $G \in \mathcal{G}^{fd}(C)$ has a weighted potential if and only if C contains only affine functions. In this case the weighted potential is an exact potential. To the best of our knowledge, our characterization establishes for the first time the existence of an exact potential function (and hence the existence of a PNE) for affine cost functions and *arbitrary* strategy sets and demands, respectively.

Second, we study weighted congestion games with *elastic* demands. Here, each player i is allowed to choose both a subset of the set of facilities and her demand d_i out of a compact set $D_i \subset \mathbb{R}_+$ of demands that are allowable for her. This congestion model can be interpreted as a generalization of Cournot games [7], where multiple producersstrategically determine quantities they will produce. The cost of a producer is given by her offered quantity multiplied with the market price, which is usually a decreasing function of the total quantity offered by all producers. Weighted congestion games with elastic demands generalize Cournot games in the sense that there are multiple markets (facilities) and each player may offer her quantity on allowable subsets of these markets.

Weighted congestion games with elastic demands have several more natural applications: they model, e.g., routing problems in the Internet, where each user wants to route data along a path in the network and adjusts the injected data rate according to the level of congestion in the network. Most mathematical models for routing and congestion control rely on fractional routing, see Kelly [17] and Cole et al. [6]. In practice, however, routing protocols use single path routing, see, e.g., the current TCP/IP protocol. Weighted congestion games with elastic demands model both congestion control and unsplittable routing. Yet another application is that of Shapley cost sharing games with players that may vary their requested demand.

Let $\mathcal{G}^e(C)$ be the set of weighted congestion games with elastic demands where each player may chose her demand out of a compact space and where the cost of each facility is determined by a function in C. Our main contribution is to show that all games $G \in \mathcal{G}^e(C)$ are weighted potential games if and only if C contains only affine functions. For this important class of games, our result also establishes for the first time the existence of PNE.

Proofs of our results can be found in [15]. In a follow up paper [16] we characterize strong Nash equilibria for weighted congestion games with bottleneck objectives.

2 Preliminaries

Definition 1 (Finite game). *A finite strategic game is a tuple $G = (N,X,\pi)$ where $N = \{1,\ldots,n\}$ is the non-empty finite set of players, $X = \bigtimes_{i\in N} X_i$ where X_i is the finite and non-empty set of strategies of player i, and $\pi : X \to \mathbb{R}^n$ is the combined private cost function.*

We will call an element $x \in X$ a strategy profile. For $S \subset N$, $-S$ denotes the complementary set of S, and we define for convenience of notation $X_S = \bigtimes_{j\in S} X_j$. Instead of $X_{-\{i\}}$ we will write X_{-i}, and with a slight abuse of notation we will write sometimes a strategy profile as $x = (x_i, x_{-i})$ meaning that $x_i \in X_i$ and $x_{-i} \in X_{-i}$.

Definition 2 (Weighted potential game, exact potential game). *A strategic game $G = (N,X,\pi)$ is called* weighted potential game *if there is a vector $w = (w_i)_{i\in N}$ of positive weights and a real-valued function $P : X \to \mathbb{R}$ such that $\pi_i(x_i, x_{-i}) - \pi_i(y_i, x_{-i}) = w_i\,(P(x_i, x_{-i}) - P(y_i, x_{-i}))$ for all players $i \in N$ and for all $x_{-i} \in X_{-i}$ and all $x_i, y_i \in X_i$. The function P together with the vector w is then called a weighted potential of the game G. The function P is called an* exact *potential if $w_i = 1$ for all $i \in N$.*

We sometimes call a weighted potential function P a $(w_i)_{i\in N}$-potential.

Monderer and Shapley [22, Theorem 2.8] have shown that one can characterize exact potentials in a very convenient way. For this, let a finite strategic game $G = (N,X,\pi)$ be given. A *path* in X is a sequence $\gamma = (x^0, x^1, \ldots x^m)$ with $x^k \in X$, $k = 0,\ldots,m$, such that for all $k \in \{1,\dots,m\}$ there exists a unique player $i_k \in N$ such that $x^k = (x^k_{i_k}, x^{k-1}_{-i_k})$ for some $x^k_{i_k} \neq x^{k-1}_{i_k}$, $x^k_{i_k} \in X_i$. A path is called closed if $x^0 = x^m$ and is called simple if $x^k \neq x^l$ for $k \neq l$. The length of a closed path is defined as the number of its distinct elements. For a set of strategy profiles X let $\Gamma(X)$ denote the set of all simple closed paths in X that have length 4. For a finite path $\gamma = (x^0, x^1, \dots, x^m)$ let the discrete path integral of π along γ be defined as $I(\gamma,\pi) = \sum_{k=1}^m \left(\pi_{i_k}(x^k) - \pi_{i_k}(x^{k-1})\right)$ where i_k is the deviator at step k in γ, that is $x^k_{i_k} \neq x^{k-1}_{i_k}$.

Theorem 1 (Monderer and Shapley). *Let $G = (N,X,\pi)$ be a finite strategic game. Then, G is an exact potential game if and only if $I(\gamma,\pi) = 0$ for all $\gamma \in \Gamma(X)$.*

In the following, we will use this characterization in order to study the existence of potentials in weighted congestion games.

3 Weighted Congestion Games

Definition 3 (Congestion model). *A tuple $\mathcal{M} = (N, F, X = \bigtimes_{i\in N} X_i, (c_f)_{f\in F})$ is called a* congestion model*, where $N = \{1,\dots,n\}$ is a non-empty, finite set of players, F is a non-empty, finite set of facilities, for each player $i \in N$, her collection of pure strategies X_i is a non-empty, finite set of subsets of F and $(c_f)_{f\in F}$ is a set of cost functions.*

In the following, we will define weighted congestion games similar to Goemans et al. [14].

Definition 4 (Weighted congestion game). *Let $\mathcal{M} = (N, F, X, (c_f)_{f\in F})$ be a congestion model and $(d_i)_{i\in N}$ be a vector of demands $d_i \in \mathbb{R}_+$. The corresponding* weighted congestion game *is the strategic game $G(\mathcal{M}) = (N, X, \pi)$, where π is defined as $\pi = \bigtimes_{i\in N} \pi_i$, $\pi_i(x) = \sum_{f\in x_i} d_i\, c_f(\ell_f(x))$ and $\ell_f(x) = \sum_{j\in N: f\in x_j} d_j$.*

We call $\ell_f(x)$ the *load* on facility f in strategy x. In case there is no confusion on the underlying congestion model, we will write G instead of $G(\mathcal{M})$.

A slightly different class of games has been considered by (among others) Fotakis et al. [11,12], Gairing et al. [13] and Mavronicolas et al. [19]. They considered games that almost coincide with Definition 4 except that the private cost of every player is not scaled by her demands. We call such games *normalized* if they comply with Definition 4 except that the private costs are defined as $\bar{\pi}_i(x) = \sum_{f\in x_i} c_f(\ell_f(x))$ for all $i \in N$.

Fotakis et al. [11] show that there are normalized weighted congestion games with $c_f(\ell) = \ell$ for all $f \in F$ that are not exact potential games. They also show that any normalized weighted congestion game with linear costs on the facilities admits a weighted potential.

We state the following trivial relations between weighted congestion games and normalized weighted congestion games: Let $G = (N,X,\pi)$ and $\bar{G} = (N,X,\bar{\pi})$ be a weighted congestion game and a normalized weighted congestion game with demands $(d_i)_{i\in N}$, respectively. Moreover, let them share the same congestion model and the same demands. Then G and $\bar{G}$ coincide in the following sense: (i) A strategy profile $x \in X$ is a PNE in G if and only if x is a PNE in $\bar{G}$; (ii) A real-valued function $P: X \to \mathbb{R}$ is a $(w_i/d_i)_{i\in N}$-potential for G if and only if P is a $(w_i)_{i\in N}$-potential for $\bar{G}$; (iii) A real-valued function

$P : X \to \mathbb{R}$ is an ordinal potential for G (see [22] for a definition) if and only if P is an ordinal potential for $\bar{G}$; (*iv*) The real-valued function $P : X \to \mathbb{R}$ is an exact potential for G if and only if P is a $(d_i)_{i\in N}$-potential for $\bar{G}$; (*v*) The real-valued function $P : X \to \mathbb{R}$ is an exact potential for $\bar{G}$ if and only if P is a $(1/d_i)_{i\in N}$-potential for G. All proofs rely on the simple observation that $\pi_i(x) = d_i \bar{\pi}_i(x)$ for all $i \in N, x \in X$.

3.1 Characterizing the Existence of an Exact Potential

In the following, we will examine necessary and sufficient conditions for a weighted congestion game G to be a potential game. The criterion in Theorem 1 states that the existence of an exact potential for $G = (N, X, \pi)$ is equivalent to the fact that $I(\gamma, \pi) = 0$ for all $\gamma \in \Gamma(X)$. In such paths, either one or two players deviate. It is easy to verify that $I(\gamma, \pi) = 0$ for all paths γ with only one deviating player. Considering a path γ with two deviating players, say i and j, each of them uses two different strategies, say $x_i, y_i \in X_i$ and $x_j, y_j \in X_j$. We denote by $z_{-\{i,j\}} \in X_{-\{i,j\}}$ the strategy profile of all players except i and j that remains constant in γ. Then, a generic path $\gamma \in \Gamma(X)$ can be written as $\gamma = ((x_i, x_j, z_{-\{i,j\}}), (y_i, x_j, z_{-\{i,j\}}), (y_i, y_j, z_{-\{i,j\}}), (x_i, y_j, z_{-\{i,j\}}), (x_i, x_j, z_{-\{i,j\}}))$. The following lemma provides an explicit formula for the calculation of $I(\gamma, \pi)$ for such a path.

Lemma 1. *Let $\mathcal{M} = (N, F, X, (c_f)_{f\in F})$ be a congestion model and $G(\mathcal{M})$ a corresponding weighted congestion game with demands $(d_i)_{i\in N}$. Moreover, let $\gamma = ((x_i, x_j, z_{-\{i,j\}}), (y_i, x_j, z_{-\{i,j\}}), (y_i, y_j, z_{-\{i,j\}})(x_i, y_j, z_{-\{i,j\}}), (x_i, x_j, z_{-\{i,j\}}))$ be an arbitrary path in $\Gamma(X)$ with two deviating players. Then,*

$$\begin{aligned} I(\gamma,\pi) = &\sum_{f\in F_1\cup F_{11}} (d_j - d_i)c_f(d_i + d_j + r_f) - d_j c_f(d_j + r_f) + d_i c_f(d_i + r_f) \\ &+ \sum_{f\in F_3\cup F_9} (d_i - d_j)c_f(d_i + d_j + r_f) - d_i c_f(d_i + r_f) + d_j c_f(d_j + r_f), \end{aligned} \tag{1}$$

where $F_1 = (x_i \setminus y_i) \cap (x_j \setminus y_j)$, $F_3 = (x_i \setminus y_i) \cap (y_j \setminus x_j)$, $F_9 = (y_i \setminus x_i) \cap (x_j \setminus y_j)$, and $F_{11} = (y_i \setminus x_i) \cap (y_j \setminus x_j)$.

Using Lemma 1, we can derive a sufficient condition on the existence of an exact potential in a weighted congestion game.

Proposition 1. *Let $\mathcal{M} = (N, F, X, (c_f)_{f\in F})$ be a congestion model and $G(\mathcal{M})$ a corresponding weighted congestion game with demands $(d_i)_{i\in N}$. For each facility $f \in F$ we denote by $N^f = \{i \in N : (\exists x_i \in X_i : f \in x_i))\}$ the set of players potentially using f, and by $\mathcal{R}^f_{-\{i,j\}} = \left\{\sum_{k\in P} d_k : P \subseteq N^f \setminus \{i,j\}\right\}$ the set of possible residual demands by all players except i and j. If for all $f \in F$ and all $i, j \in N^f$ it holds that*

$$(d_j - d_i)c_f(d_i + d_j + r_f) - d_j c_f(d_j + r_f) + d_i c_f(d_i + r_f) = 0 \quad \forall r_f \in \mathcal{R}^f_{-\{i,j\}}, \tag{2}$$

then G admits an exact potential.

It is a useful observation that we can write the condition of Proposition 1 as

$$\frac{c_f(d_i + d_j + r_f) - c_f(d_j + r_f)}{d_i} = \frac{c_f(d_j + r_f) - c_f(d_i + r_f)}{d_j - d_i} \tag{3}$$

for all $i,j \in N^f$ and $r_f \in \mathcal{R}^f_{-\{i,j\}}$. Thus, the difference quotients of c_f between the points $d_i + r_f$ and $d_j + r_f$ as well as $d_j + r_f$ and $d_i + d_j + r_f$ must have the same value. It follows easily that the above condition is satisfied if all demands are equal (this corresponds to unweighted congestion games, see Rosenthal's potential [24]). For *arbitrary* demands (weighted congestion games) and *affine* cost functions, one can check that the above condition is also satisfied, see the positive result of Fotakis et al. [11].

There is, however, an important question left: Are there non-affine cost functions that give rise to an exact potential in *all* weighted congestion games? Under mild assumptions on feasible cost functions, we will give in Theorem 2 a negative answer to this question. First, we derive the following lemma from Theorem 1.

Lemma 2. *Let C be a set of functions and let $\mathcal{G}(C)$ be the set of all weighted congestion games with cost functions in C. Every $G \in \mathcal{G}(C)$ has an exact potential if and only if for all $c \in C$*

$$(x-y)\,c(x+y+z) - x\,c(x+z) + y\,c(y+z) = 0 \tag{4}$$

for all $x,y \in \mathbb{R}_+$ and $z \in \mathbb{R}^0_+$.

We will now solve the functional equation (4) in order to characterize all cost functions that guarantee an exact potential in all weighted congestion games. We require the following property: A function $c : \mathbb{R}_+ \to \mathbb{R}$ is *locally bounded*, if for every compact set $K \subset \mathbb{R}_+$, $|c(x)| < M_K$ for all $x \in K$ and a constant $M_K \in \mathbb{R}_+$ potentially depending on K.

Theorem 2. *Let C be a set of locally bounded functions and let $\mathcal{G}(C)$ be the set of weighted congestion games with cost functions in C. Then every $G \in \mathcal{G}(C)$ admits an exact potential function if and only if C contains affine functions only, that is, every $c \in C$ can be written as $c(\ell) = a_c\,\ell + b_c$ for some $a_c, b_c \in \mathbb{R}$.*

3.2 Characterizing the Existence of a Weighted Potential

Our next aim is to determine whether weaker notions of potential functions will enrich the class of cost functions giving rise to a potential game. The idea of a weighted potential allows a player specific scaling of the private cost π_i by a strictly positive w_i. It is a useful observation that the existence of a weighted potential function is equivalent to the existence of a strictly positive-valued vector $w = (w_i)_{i \in N}$ such that the game G^w with private costs $\bar{\pi} := \bigtimes_{i \in N} \pi_i / w_i$ has an exact potential.

Using this equivalent formulation and Theorem 1 it follows that the existence of an exact potential function for the game $G^w = (N, X, \bar{\pi})$ is equivalent to $I(\gamma, \bar{\pi}) = 0$ for all $\gamma \in \Gamma(X)$ suggesting that G^w has an exact potential if and only if there are $w_i, w_j \in \mathbb{R}_+$ such that

$$\left(\frac{d_i}{w_i} - \frac{d_j}{w_j}\right) c_f(d_i + d_j + r_f) = \frac{d_i}{w_i} c_f(d_i + r_f) - \frac{d_j}{w_j} c_f(d_j + r_f)$$

for all $i, j \in N$ and all $r_f \in \mathcal{R}_{-i,j}$. In particular it is necessary that either $c_f(d_i + d_j + r_f) = c_f(d_j + r_f) = c_f(d_i + r_f)$ or the value $\alpha(d_i, d_j)$ defined as

$$\alpha(d_i, d_j) = \frac{w_i}{w_j} = \frac{d_i}{d_j} \cdot \frac{c_f(d_i + d_j + r_f) - c_f(d_i + r_f)}{c_f(d_i + d_j + r_f) - c_f(d_j + r_f)} \tag{5}$$

is strictly positive and independent of both f and r_f.

Lemma 3. *Let C be a set of functions. Let $\mathcal{G}(C)$ be the set of weighted congestion games with cost functions in C. Every $G \in \mathcal{G}(C)$ has a weighted potential if and only if for all $x, y \in \mathbb{R}_+$ there exists an $\alpha(x,y) \in \mathbb{R}_+$ such that*

$$\alpha(x,y) = \frac{x}{y} \cdot \frac{c(x+y+z) - c(x+z)}{c(x+y+z) - c(y+z)} \tag{6}$$

for all $z \in \mathbb{R}_+^0$ and non-constant $c \in C$.

Although this condition seems to be similar to the functional equation (4) characterizing the existence of an exact potential, it is not possible to proceed using differential equations as in the proof of Theorem 2. As $\alpha(x,y)$ need not be bounded it is not possible to prove continuity and differentiability of c. Instead, we will use the discrete counterpart of differential equations, that is, difference equations.

Theorem 3. *Let C be a set of continuous functions. Let $\mathcal{G}(C)$ be the set of weighted congestion games with cost functions in C. Then every $G \in \mathcal{G}(C)$ admits a weighted potential if and only if exactly one of the following cases holds:*

1. *C contains only affine functions,*
2. *C contains only exponential functions $c(\ell) = a_c e^{\phi\ell} + b_c$ for some $a_c, b_c, \phi \in \mathbb{R}$, where a_c and b_c may depend on c, while ϕ must be equal for every $c \in C$.*

3.3 Implications of Our Characterizations

It is natural to ask whether these results remain valid if additional restrictions on the set $\mathcal{G}(C)$ are made. A natural restriction is to assume that all players have an integral demand. As we used infinitesimally small demands in the proof of Lemma 2, our results for exact potentials do not apply directly to integer demands. With a slight variation of the proof of Theorem 3 where only the case $\alpha(\cdot,\cdot) = 1$ is considered, however, we still obtain the same result provided that C contains only continuous functions.

Another natural restriction on $\mathcal{G}(C)$ are games with symmetric sets of strategies or games with a bounded number of players or facilities. Since the proofs of Lemma 2 and 3 and Theorems 2 and 3 rely on mild assumptions, we can strengthen our characterizations as follows.

Corollary 1. *Let C be a set of continuous functions. Let $\mathcal{G}(C)$ be the set of weighted congestion games with cost functions in C satisfying one or more of the following properties*

1. *Each game $G = (N, X, \pi) \in \mathcal{G}(C)$ has two (three) players.*
2. *Each game $G = (N, X, \pi) \in \mathcal{G}(C)$ has three (five) facilities.*
3. *For each game $G = (N, X, \pi) \in \mathcal{G}(C)$ and each player $i \in N$ the set of her strategies X_i contains a single facility only.*
4. *Each game $G = (N, X, \pi) \in \mathcal{G}(C)$ has symmetric strategies, that is $X_i = X_j$ for all $i, j \in N$.*
5. *In each game $G = (N, X, \pi) \in \mathcal{G}(C)$ the demands of all players are integral.*

Then, every $G = (N,X,\pi) \in \mathcal{G}(C)$ has an exact (a weighted) potential if and only if C contains only affine functions (only affine functions or only exponential functions as in Theorem 3).

Yet, we are able to deduce an interesting result concerning the existence of weighted potentials in weighted congestion games where each facility can be chosen by at most two players. As we can set $z = 0$ in (6), the conditions of Lemma 3 are fulfilled by more than affine or exponential functions.

Theorem 4. *Let $m(x)$ be a strictly monotonic function and $C_m = \{a\,m(x) + b : a, b \in \mathbb{R}\}$. Let $\mathcal{G}^2(C_m)$ be the set of games such that cost functions are in C_m and every facility is contained in the set of strategies of at most two players. Then, every $G \in \mathcal{G}^2(C_m)$ possesses a weighted potential.*

This result generalizes a result of Anshelevich et al. in [3], who showed that a weighted congestion game with two players and $c_f(\ell) = b_f/\ell$ for a constant $b_f \in \mathbb{R}_+$ has a potential. Moreover, this result shows that the characterization of Corollary 1 is tight in the sense that weighted congestion games with two players admit a weighted potential even if cost functions are neither affine nor exponential.

4 Extensions of the Model

In the last section, we developed a new technique to characterize the set of functions that give rise to a potential in weighted congestion games. In this section, we will introduce two generalizations of weighted congestion games and investigate the set of cost functions that assure the existence of potential functions.

Definition 5 (Weighted congestion game with facility-dependent demands). *Let $\mathcal{M} = (N, F, X, (c_f)_{f\in F})$ be a congestion model and let $\left(d_{i,f}\right)_{i\in N, f\in F}$ be a matrix of facility-dependent demands. The corresponding* weighted congestion game with facility-dependent demands *is the strategic game $G(\mathcal{M}) = (N, X, u)$, where u is defined as $u = \bigtimes_{i\in N} \pi_i$, $\pi_i(x) = \sum_{f\in x_i} d_{i,f}\, c_f(\ell_f(x))$ and $\ell_f(x) = \sum_{j\in N: f\in x_j} d_{j,f}$.*

Restricting the strategy sets to singletons, we obtain scheduling games. In a scheduling game, players are jobs that have machine-dependent demands and can be scheduled on a set of admissible machines (restricted scheduling on unrelated machines). In contrast to the classical approach, where each job strives to minimize its makespan, we consider a different private cost function: Machines charge a price per unit given by a load-dependent cost function c_f and each job minimizes its cost defined as the price of the chosen machine multiplied with its machine-dependent demand.

Theorem 5. *Let C be a set of continuous functions and let $\mathcal{G}^{fd}(C)$ be the set of weighted congestion games with facility-dependent demands and cost functions in C. Then, every $G \in \mathcal{G}^{fd}(C)$ admits a weighted potential if and only if C contains only affine functions, that is, every $c \in C$ can be written as $c(\ell) = a_c\,\ell + b_c$ for some $a_c, b_c \in \mathbb{R}$. For a game G with affine cost functions, the potential function is given by $P(x) = \sum_{i\in N} \sum_{f\in x_i} c_f\left(\sum_{j\in\{1,\dots,i\}: f\in x_j} d_{j,f}\right) d_{i,f}$.*

We will now introduce an extension to weighted congestion games allowing players to also choose their demand.

Definition 6 (Weighted congestion game with elastic demands). *Let* $\mathcal{M} = (N, F, X, (c_f)_{f\in F})$ *be a congestion model. Together with* $D = \bigtimes_{i\in N} D_i$*, where* $D_i \subset \mathbb{R}_+$ *are compact for all* $i \in N$*, we define the* weighted congestion game with elastic demands *as the strategic game* $G(\mathcal{M}) = (N, \bar{X}, \pi)$ *with* $\bar{X} := (X, D)$*,* $\pi = \bigtimes_{i\in N} \pi_i$*, and* $\pi_i(\bar{x}) = \sum_{f\in x_i} d_i c_f(\ell_f(\bar{x}))$ *and* $\ell_f(\bar{x}) = \sum_{j\in N: f\in x_j} d_j$*.*

In our definition of weighted congestion games with elastic demands, we explicitly allow for positive and negative, and for increasing and decreasing cost functions. Thus, an increase in the demand may increase or decrease the player's private cost. Note that in weighted congestion games with elastic demands, the strategy sets are topological spaces and are in general infinite. By restricting the sets D_i to singletons $D_i = \{d_i\}, i \in N$, we obtain weighted congestion games as a special case of weighted congestion games with elastic demands. The proof of the following result is similar to the case of facility-dependent demands.

Theorem 6. *Let* C *be a set of continuous functions and let* $\mathcal{G}^e(C)$ *be the set of weighted congestion games with elastic demands and cost functions in* C*. Then, eve-ry* $G \in \mathcal{G}^e(C)$ *admits a weighted potential function if and only if* C *contains only affine functions. For a game* G *with affine cost functions, the potential function is given by* $P(\bar{x}) = \sum_{i\in N} \sum_{f\in x_i} c_f\left(\sum_{j\in\{1,\dots,i\}: f\in x_j} d_j\right) d_i$*.*

As an immediate consequence, we obtain the existence of a PNE if cost functions are affine. Note that the existence of a potential is not sufficient for proving existence of a PNE as we are considering infinite games. However, as $\bar{X}$ is compact and P is continuous, P has a minimum $\bar{x}^* \in \bar{X}$ and $\bar{x}^*$ is a PNE.

Corollary 2. *Let* C *be a set of affine functions and let* $\mathcal{G}^e(C)$ *be the set of weighted congestion games with elastic demands and cost functions in* C*. Then every* $G \in \mathcal{G}^e(C)$ *admits a PNE.*

References

1. Ackermann, H., Röglin, H., Vöcking, B.: Pure Nash equilibria in player-specific and weighted congestion games. Theor. Comput. Sci. 410(17), 1552–1563 (2009)
2. Albers, S., Eilts, S., Even-Dar, E., Mansour, Y., Roditty, L.: On Nash equilibria for a network creation game. In: Proc. of the 17th Symposium on Discrete Algorithms (SODA), pp. 89–98 (2006)
3. Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, E., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. In: Proc. of the 45th Symposium on Foundations of Computer Science (FOCS), pp. 295–304 (2004)
4. Awerbuch, B., Azar, Y., Epstein, A., Mirrokni, V.S., Skopalik, A.: Fast convergence to nearly optimal solutions in potential games. In: Proc. of the 9th conference on Electronic commerce (EC), pp. 264–273 (2008)
5. Chen, H.-L., Roughgarden, T.: Network design with weighted players. In: Proc. of the 18th Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 29–38 (2006)

6. Cole, R., Dodis, Y., Roughgarden, T.: Bottleneck links, variable demand, and the tragedy of the commons. In: Proc. of the 17th Symposium on Discrete Algorithms (SODA), pp. 668–677 (2006)
7. Cournot, A.G.: Recherches Sur Les Principes Mathematiques De La Theorie De La Richesse. Hachette, Paris (1838)
8. Even-Dar, E., Kesselman, A., Mansour, Y.: Convergence time to Nash equilibrium in load balancing. ACM Trans. Algorithms 3(3), 32 (2007)
9. Fabrikant, A., Papadimitriou, C., Talwar, K.: The complexity of pure Nash equilibria. In: Proc. of the 36th Symposium on Theory of Computing (STOC), pp. 604–612 (2004)
10. Fotakis, D., Kaporis, A.C., Spirakis, P.G.: Atomic congestion games: Fast, myopic and concurrent. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 121–132. Springer, Heidelberg (2008)
11. Fotakis, D., Kontogiannis, S., Spirakis, P.: Selfish unsplittable flows. Theoretical Computer Science 348(2-3), 226–239 (2005)
12. Fotakis, D., Kontogiannis, S., Spirakis, P.: Atomic congestion games among coalitions. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 572–583. Springer, Heidelberg (2006)
13. Gairing, M., Monien, B., Tiemann, K.: Routing (un-) splittable flow in games with player-specific linear latency functions. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 501–512. Springer, Heidelberg (2006)
14. Goemans, M., Mirrokni, V., Vetta, A.: Sink equilibria and convergence. In: Proc. of the 46th Symposium on Foundations of Computer Science (FOCS), pp. 142–154 (2005)
15. Harks, T., Klimm, M., Möhring, R.H.: Characterizing the existence of potential functions in weighted congestion games. Technical Report 11, COGA, TU Berlin (2009)
16. Harks, T., Klimm, M., Möhring, R.H.: Strong Nash equilibria in games with the lexicographical improvement property. Technical Report 17, COGA, TU Berlin (2009)
17. Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate Control in Communication Networks: Shadow Prices, Proportional Fairness, and Stability. Journal of the Operational Research Society 49, 237–252 (1998)
18. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. Telecommunication Systems 17(4), 385–409 (2001)
19. Mavronicolas, M., Milchtaich, I., Monien, B., Tiemann, K.: Congestion games with player-specific constants. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 633–644. Springer, Heidelberg (2007)
20. Milchtaich, I.: Congestion games with player-specific payoff functions. Games and Economic Behavior 13(1), 111–124 (1996)
21. Monderer, D., Shapley, L.S.: Fictitious play property for games with identical interests. Journal of Economic Theory 68(1), 258–265 (1996)
22. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14(1), 124–143 (1996)
23. Panagopoulou, P.N., Spirakis, P.G.: Algorithms for pure Nash equilibria in weighted congestion games. Journal on Experimental Algorithmics 11, 2–7 (2006)
24. Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. International Journal of Game Theory 2(1), 65–67 (1973)
25. Vöcking, B.: Selfish load balancing. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (eds.) Algorithmic Game Theory, pp. 517–542. Cambridge University Press, Cambridge (2007)

Free-Riding and Free-Labor in Combinatorial Agency

Moshe Babaioff[1], Michal Feldman[2], and Noam Nisan[3]

[1] Microsoft Research - Silicon Valley
moshe@microsoft.com
[2] School of Business Administration, Hebrew University of Jerusalem
michal.feldman@huji.ac.il
[3] School of Computer Science, Hebrew University of Jerusalem
noam@cs.huji.ac.il

Abstract. This paper studies a setting where a principal needs to motivate teams of agents whose efforts lead to an outcome that stochastically depends on the combination of agents' actions, which are not directly observable by the principal. In [1] we suggest and study a basic "combinatorial agency" model for this setting. In this paper we expose a somewhat surprising phenomenon found in this setting: cases where the principal can gain by asking agents to *reduce* their effort level, even when this increased effort comes *for free*. This phenomenon cannot occur in a setting where the principal can observe the agents' actions, but we show that it can occur in the hidden-actions setting. We prove that for the family of technologies that exhibit "increasing returns to scale" this phenomenon cannot happen, and that in some sense this is a maximal family of technologies for which the phenomenon cannot occur. Finally, we relate our results to a basic question in production design in firms.

1 Introduction

Background: Combinatorial Agency

The well studied principal-agent problem deals with how a "principal" can motivate a rational "agent" to exert costly effort towards the welfare of the principal. The difficulty in this model is that the agent's action (i.e. whether he exerts effort or not) is unobservable by the principal and only the final outcome, which is probabilistic and also influenced by other factors, is observable. "Unobservable" here is meant in a wide sense that includes "not precisely measurable", "costly to determine", or "non-contractible" (meaning that it can not be upheld in "a court of law"). This problem is well studied in many contexts in classical economic theory and we refer the reader to introductory texts on economic theory for background (e.g. [11] Chapter 14). The solution is based on the observation that a properly designed contract, in which the payments are contingent upon the final outcome, can influence a rational agent to exert the required effort.

In [1] we initiated a general study of handling *combinations* of agents rather than a single agent. While much work was previously done on motivating teams

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 109–121, 2009.

of agents [8,13,9,3], our emphasis in [1] was on dealing with the complex combinatorial structure of dependencies between agents' actions.

In the general model presented in [1], each of n agents has a set of possible *actions*, the combination of actions by the players results in some *outcome*, where this happens probabilistically. The main part of the specification of a problem in this model is a function that specifies this distribution for each n-tuple of agents' actions ("the technology"). Additionally, the problem specifies the principal's utility for each possible outcome, and for each agent, the agent's cost for each possible action. The principal motivates the agents by offering to each of them a *contract* that specifies a payment for each possible outcome of the whole project, with the goal of maximizing his expected net utility. Key here is that the actions of the players are non-observable and thus the contract cannot make the payments directly contingent on the actions of the players, but rather only on the outcome of the whole project.

Given a set of contracts, the agents will each optimize his own utility; i.e., will choose the action that maximizes his expected payment minus the cost of the action. Since the outcome depends on the actions of all players together, the agents are put in a game here and are assumed to reach a Nash Equilibrium (NE). The principal's problem is that of designing the *optimal contract*: i.e. the vector of contracts to the different agents that induce an equilibrium that will optimize his expected utility from the outcome minus his expected total payment.

We refer the reader to our earlier paper [1] for further motivation and details. Several other papers study different issues in the combinatorial agency model. Mixed strategies were studied in [2], while [6] studied random audits. In this paper we deal with a rather surprising (to us) phenomena that we have discovered in this model: the possible advantage of "throwing away" some free agents' effort (effort increase with no increase in cost).

Our Results

We focus on the case of two possible outcomes ("binary outcome"): either the project succeeds (generating value v to the principal) or fails (value 0). We generalize the model of [1] and allow for more than two actions for each agent. In this multiple-actions setting it is natural to assume that each agent has a linear order over his actions that corresponds to the actions' cost, and that more effort (according to the linear order) does not decrease the project's probability of success. An agent *wastes free labor* if he plays an action for which there exists another action with the same cost and is better according the linear order (as the project's success probability can increase with no increase in the agent's cost). A contract wastes free labor if at least one of the agents plays an action that wastes free labor. Is it possible that the principal's optimal contract will waste free labor? In the observable-actions case the principal can never gain by such a waste. Somewhat surprisingly, in the hidden-actions case we are able to present an example for which the principal can gain by wasting free-labor (Section 3). The fundamental reason for that is that free labor increases *free riding*, and reduces the motivation of other agents to exert effort.

To measure the principal's loss from using free labor we define the *Price of Free-Labor (POFL)*. POFL is defined to be the worse ratio (over all values v) between the principal's utility in the optimal contract and the best contract that must use all free labor (Section 4). Our goal is to characterize technologies for which free labor is never wasted. We show that for technologies that exhibit "increasing returns to scale (IRS)", free labor is never wasted (Section 5). Informally, the IRS property ensures that an increase in effort of all agents but one increases the marginal contribution of that agent due to an increase in his effort. An example for such a technology is the AND technology in which agents are perfect complements, each agent has a sub-task and the project succeeds only if all agents succeed in their sub-tasks. Thus, the IRS condition is sufficient to ensure that free labor is not wasted. Is it also necessary? It is easy to construct arbitrary technologies that do not exhibit IRS yet do not waste free labor.[1] Therefore we focus on a natural and large family of technologies: "structured technologies", and aim to prove a complementary result for a natural form of free labor in that family.

In a "structured technology" each agent has a sub-task to perform and the project's success is a deterministic Boolean function of the set of successful sub-tasks. The success of a sub-task executed by an agent is determined independently and stochastically as a function of his effort. If he exert no effort he bears no cost and the success probability is low, while if he exerts effort the cost is positive and the success probability is higher. Free labor is introduced by the principal's ability to remove agents altogether. Suppose that a given technology function specifies the underlying technological feasibility, but now the range of possible technologies that the principal can apriori choose among is given by the *sub-technologies* of the given one. I.e. the principal can apriori choose a subset of the agents and completely removing the others – in which case all the subtasks of the removed agents will surely fail.[2] Removed agents as well as agents that do not exert effort bear no cost. If an agent supply some positive success probability for his sub-task without any effort then removing the agent corresponds to a waste of free labor.

We ask the following question: for which technology functions a waste of free labor will never occur (independent of the exact parameters of the agents' success probabilities in their sub-tasks) ? We show that *any* structured technology will waste free labor for some choice of parameters, with a single exception: for the AND function, with any choice of parameters, free labor should always be used.

Finally, we draw a connection between this phenomenon and the much discussed question of process-based (PB) vs. function-based (FB) division of labor [10,12,14]: Suppose that a firm produces a product (task) that is composed of two parts (sub-tasks): A and B. Two workers (agents) $A1$ and $A2$ can each perform a sub-task of type A and two other workers $B1$ and $B2$ can each perform a sub-task of type B. One can consider two natural ways of organizing the production in the firm:

[1] Actually, if each action has a different cost this holds trivially.

[2] One could assign different costs to the different sub-technologies, but we just look at the simplest question without any associated costs.

- **Function based:** Two “divisions”, each consisting of one agent of *each* type. The project succeeds if at least one division is successful. The success here can be represented by ($A1$ AND $B1$) OR ($A2$ AND $B2$).
- **Process based:** Two “divisions”, each consisting of two agents of the *same* type. In this case there is an “A division” (with $A1, A2$) and a “B division” (with $B1, B2$). The success here can be represented by ($A1$ OR $A2$) AND ($B1$ or $B2$).

Notice that the process-based organization is superior in terms of probability of success: the function-based alternative simply discards the possibility of success due to ($A1$ AND $B2$) OR ($A2$ AND $B1$). Yet, our results show that in an agent-based setting with hidden actions, the function-based approach may still be superior due to lower level of possible free-riding. We discuss the connection to the issue of free labor at Section 7. This result seems to be in line with the main intuitive reasons for choosing function-based organization (see [14]).

Due to lack of space we defer all proofs to the full version of the paper (which can be found on the authors' web sites).

2 Model and Preliminaries

Our main interest is in the simple “binary action, binary outcome” scenario where each agent has two possible actions (”exert effort” or ”shirk”) and there are two possible outcomes (”failure”, ”success”). In order to study phenomena in this setting, we will need to work within a more general model in which agents have general actions, but the outcome is still binary. This falls within the general framework of [1], and generalizes the “binary action” sub-model.

A principal employs a set of agents N of size n. Each agent $i \in N$ has a possible set of actions A_i, and a cost (effort) $c_i(a_i) \geq 0$ for each possible action $a_i \in A_i$ ($c_i : A_i \rightarrow \Re_+$). The actions of all players determine, in a probabilistic way, a “contractible” outcome, $o \in \{0, 1\}$, where the outcomes 0 and 1 denote project failure and success, respectively (binary-outcome). The outcome is determined according to a success function $t : A_1 \times \ldots \times A_n \rightarrow [0, 1]$, where $t(a_1, \ldots, a_n)$ denotes the probability of project success where players play with the action profile $a = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n = A$. We use the notation $(t, c(\cdot))$ to denote a technology (a success function and a cost function for each agent).

The principal's value of a successful project is given by a scalar $v > 0$, where he gains no value from a project failure. The idea is that the actions of the players are unobservable, but the final outcome o is observed by him and others, and he may design enforceable contracts based on this outcome. We assume that the principal can pay the agents but not fine them (known as the *limited liability* constraint). The contract to agent i is thus given by a scalar value $p_i \geq 0$ that denotes the payment that i gets in case of project success. If the project fails, the agent gets no money.

Given this setting, the agents have been put in a game, where the utility of agent i under the profile of actions $a = (a_1, \ldots, a_n)$ is given by

$u_i(a) = p_i \cdot t(a) - c_i(a_i)$. As usual, we denote by $a_{-i} \in A_{-i}$ the $(n-1)$-dimensional vector of the actions of all agents excluding agent i. i.e., $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. The agents will be assumed to reach Nash equilibrium, if such an equilibrium exists. The principal's likes to design the contracts p_i as to maximize his own expected utility $u(a, v) = t(a) \cdot (v - \sum_{i \in N} p_i)$, where the actions $a_1, \ldots, a_n$ are at Nash-equilibrium. In the case of multiple Nash equilibria, in our model we let the principal choose the desired one, and "suggest" it to the agents, thus focusing on the "best" Nash equilibrium.[3]

As we wish to concentrate on motivating agents, rather than on the coordination between agents, we assume that more effort by an agent always leads to a better probability of success. Formally, we assume that the actions of each agent are ordered according to the amount of effort, i.e. for any i there is a linear order $\succ_i$ on A_i that is consistent with the costs, $a_i \succ_i a'_i \Rightarrow c_i(a_i) \geq c_i(a'_i)$, and the success function t is monotone non-decreasing, $\forall i \in N, \forall a_{-i} \in A_{-i}$ we have that $a_i \succ_i a'_i \Rightarrow t(a_i, a_{-i}) \geq t(a'_i, a_{-i})$. We also assume that $t(a) > 0$ for any $a \in A$. We denote $a_i \succeq_i a'_i$ if $a_i \succ_i a'_i$ or $a_i = a'_i$.

We start with the characterization of Nash equilibrium in this setting.

Observation 1. *The profile of actions $a \in A$ is a Nash equilibrium[4] under the payments $(p_1, p_2, \ldots, p_n)$ (agent i is paid $p_i \geq 0$ if the project succeeds and 0 if not) if and only if for any agent $i \in N$ the payment p_i satisfies[5]*

$$max_{a'_i \prec_i a_i} \frac{c_i(a_i) - c_i(a'_i)}{t_i(a_i, a_{-i}) - t_i(a'_i, a_{-i})} \leq p_i \leq min_{a'_i \succ_i a_i} \frac{c_i(a'_i) - c_i(a_i)}{t_i(a'_i, a_{-i}) - t_i(a_i, a_{-i})}$$

Moreover, to get the lowest cost payments that induce $a \in A$ as a Nash equilibrium, the lower bound weak inequality must hold as equality.

Given the technology and the value v of the principal from a successful project, the principal's goal is to maximize his utility, i.e. to determine a profile of actions $a \in A$, which gives the highest utility $u(a, v)$ in equilibrium, as calculated above. We call a profile of actions $a \in A$ that maximizes the principal's utility for the value v, an *optimal contract* for v. A simple but crucial observation, generalizing a similar one in [1], shows that the optimal contract exhibits some monotonicity properties in the value.

---

[3] A variant, which is similar in spirit to "strong implementation" in mechanism design, and discussed here, would be to take the worst Nash equilibrium, or even, stronger yet, to require that only a single equilibrium exists.

[4] Note that, unlike in the Boolean action case studied in [1], it is possible that some profile of actions cannot be a Nash equilibrium with any payments, as no payments satisfy all these conditions.

[5] If t is not strictly monotone, it might be that for some a'_i it holds that $t_i(a_i, a_{-i}) = t_i(a'_i, a_{-i})$. In this case for $a \in A$ to be a NE, it must be the case that $c_i(a'_i) \geq c_i(a_i)$. In this case we interpret the above conditions as follows. The upper bound inequality holds for any $p_i \geq 0$ (as $c_i(a'_i) \geq c_i(a_i)$ for any $a'_i \succ_i a_i$). The lower bound inequality holds if for $a'_i \prec_i a_i$, $c_i(a'_i) = c_i(a_i)$.

Lemma 1. (**Monotonicity lemma**)*: For any technology $(t, c(\cdot))$ the expected utility of the principal at the optimal contracts, the success probability of the optimal contracts, and the expected payment of the optimal contract, are all monotonically non-decreasing with the principal's value v.*

A similar lemma also holds in the observable-actions case, and is also showed there.

The principal can determine the action profile played by the agents in equilibrium by changing the agents' contracts (payment in case of success). As the value of v increases, the principal may change the profile of actions obtained at the equilibrium. It turn out that it is helpful to look at values v in which there is a change in the contracted action profile, and we call such points (values) "transition points".

Definition 1. *$v \in \Re_+$ is a* transition point *for technology $(t, c(\cdot))$ if for any $\epsilon_1 > 0$ and $\epsilon_2 > 0$, the set of optimal contracts for the value $v - \epsilon_1$ is different from the set of optimal contracts for the value $v + \epsilon_2$.*

Some of the results in this paper will be related to success functions for which the marginal contribution of any agent is non-decreasing in the effort of the other agents, we say that in such a case the function exhibits increasing returns to scale. Formally, for two action profiles $a, b \in A$ we denote $b \succeq a$ if for all j, $b_j \succeq_j a_j$.

Definition 2. *A technology success function t exhibits (weakly)* **increasing returns to scale (IRS)** *if for every i, and every $b \succeq a$*

$$t_i(b_i, b_{-i}) - t_i(a_i, b_{-i}) \geq t_i(b_i, a_{-i}) - t_i(a_i, a_{-i})$$

If a technology success function exhibits IRS we also say that the technology exhibits IRS.

2.1 Structured Technology Functions

Much of our focus will be on technology functions whose structure can be described easily as being derived from independent agent sub-tasks – called *structured technology functions*. This subclass will first give us some natural examples of technology functions, and will also provide a succinct and natural way to represent technology success functions.

In a structured technology function, each individual succeeds or fails in his own "sub-task" independently. The project's success or failure deterministically depends, maybe in a complex way, on the set of successful sub-tasks. Thus we will assume a monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which denotes whether the project succeeds as a function of the success of the n agents' tasks.

A model with a structured technology success function is a special case of the *binary-outcome, binary-action model* [1]. In this model, the action space of each agent has two possible actions: 0 (shirk) and 1 (exert effort). The cost of shirking is 0, while the cost of exerting effort is $c_i > 0$.

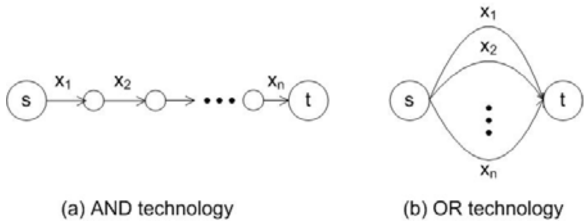


Fig. 1. Graphical representations of (a) *AND* and (b) *OR* technologies

A structure technology function t is defined by $t(a_1, \ldots, a_n)$ being the probability that $f(x_1, \ldots, x_n) = 1$ where the bits $x_1, \ldots, x_n$ are chosen according to the following distribution: if $a_i = 0$ then $x_i = 1$ with probability $\gamma_i \in [0, 1)$ (and $x_i = 0$ with probability $1 - \gamma_i$); otherwise, i.e. if $a_i = 1$, then $x_i = 1$ with probability $\delta_i > \gamma_i$ (and $x_i = 0$ with probability $1 - \delta_i$). We denote $x = (x_1, \ldots, x_n)$.

The question of the representation of the technology function is now reduced to that of representing the underlying monotone Boolean function f. In the most general case, the function f can be given by a general monotone Boolean circuit. An especially natural sub-class of functions in the structured technologies setting would be functions that can be represented as a *read-once network* – a graph with a given source and sink, where every edge is labeled by a different player. The project succeeds if the edges that belong to player's whose task succeeded form a path between the source and the sink[6].

A few simple examples should be in order here:

1. The "AND" technology: $f(x_1, \ldots, x_n)$ is the logical conjunction of x_i ($f(x) = \bigwedge_{i \in N} x_i$). Thus the project succeeds only if all agents succeed in their tasks. This is shown graphically as a read-once network in Figure 1(a).
2. The "OR" technology: $f(x_1, \ldots, x_n)$ is the logical disjunction of x_i ($f(x) = \bigvee_{i \in N} x_i$). Thus the project succeeds if at least one of the agents succeed in their tasks. This is shown graphically as a read-once network in Figure 1(b).
3. The "Or-of-Ands" (OOA) technology: $f(x_1, \ldots, x_n)$ is the logical disjunction of conjunctions. Thus the project succeeds if in at least one clause all agents succeed in their tasks. This is shown graphically as a read-once network in Figure 2(a) The simplest case is the one in which there are n_c clauses, each of length n_l; $n = n_c \cdot n_l$ (thus $f(x) = \bigvee_{j=1}^{n_c} (\bigwedge_{k=1}^{n_l} x_k^j)$).
4. The "And-of-Ors" (AOO) technology: $f(x_1, \ldots, x_n)$ is the logical conjunction of disjunctions. Thus the project succeeds if at least one agent from each disjunctive-form-clause succeeds in his tasks. This is shown graphically as a read-once network in Figure 2(b) The simplest case is the one in which there are n_c clauses of equal length n_l (thus $f(x) = \bigwedge_{j=1}^{n_l} (\bigvee_{k=1}^{n_c} x_k^j)$).

[6] One may view this representation as directly corresponding to the project of delivering a message from the source to the sink in a real network of computers, with the edges being controlled by selfish agents.

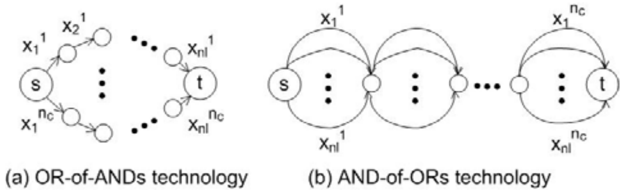


Fig. 2. Graphical representations of (a) OOA and (b) AOO technologies

A success function t is called *anonymous* if it is symmetric with respect to the players. I.e. $t(a_1, \ldots, a_n)$ depends only on $\sum_i a_i$. A technology (t, c) is *anonymous* if t is anonymous and the cost c is identical to all agents (there exists a c such that for any agent i, $c_i = c$). Of the examples presented above, if we assume that the cost c is identical to all agents and that there exists a γ such that for any agent i, $\gamma_i = 1 - \delta_i = \gamma$, then the AND and OR technologies are anonymous (while for $n_l, n_c \geq 2$, the AOO and OOA technologies are not anonymous).

2.2 Sub-technologies

The model of structured technologies (from [1]) presented above assumes that the technology function is exogenously given. In this paper we wish to ask how would the principal choose the technology function, had he had control over it. Obviously, this question is not interesting in its unrestricted form since the principal will always choose a technology in which all agents succeed with probability 1 with no cost. Yet, this question turns out to be interesting under reasonable restrictions, and it also connected to the issue of free labor. In this paper we suggest to study the "removal" model in which the principal is allowed to remove an agent, thus ensuring he will *certainly* fail in his sub-task (instead of succeeding with low probability γ_i). It seems natural to assume that if an agent is removed his cost of action is still 0.

In the "removal" model, we formally introduce the possibility of removing an agent as follows. We change the set of actions of any agent i to be $A_i = \{\emptyset, 0, 1\}$, with $1 \succ_i 0 \succ_i \emptyset$. The additional $\emptyset$ action is the action for which the agent does not participate ("removed"), and has 0 cost. If agent i is removed ($a_i = \emptyset$ and the cost to i is 0) his task will always fail, that is, $x_i = 1$ with probability 0. By removing the set of agents S the principal essentially fixes $x_i = 0$ for any $i \in S$, and this creates a restricted Boolean function $f^{|S=0}$ on the bits of the rest of the agents (S^c). We call such a restricted function a *sub-technology*.

In terms of the graphical representation as a read-once network as in Figures 1, this simply means that we allow the principal to erase, ex-ante, some of the edges. Equivalently, originally, choosing the right subset of agents to contract with was determining which agent i succeeds with probability δ_i, where the others succeeded with probability γ_i. Now, the principal can decide, within the group of non-contracted agents, a group that succeed with probability 0 rather than γ_i.

Note that by not removing an agent (and not contracting with him) the principal essentially get some "free labor" as with the same cost of 0 he get an increase in success probability. Observe that this model introduces free labor *only* for the lowest cost (0 cost) actions. This is so as for any strictly monotonic technology, it is impossible to induce Nash equilibrium in which an agent chooses a non-zero cost action that wastes his free labor (By Observation 1 if $a \in A$ is a Nash Equilibrium under p, and $a'_i \succ_i a_i$ with $t_i(a'_i, a_{-i}) > t_i(a_i, a_{-i})$ then it must be the case that $c_i(a'_i) > c_i(a_i)$.)

3 Free-Labor Might Be Costly: An Example

For the "removal" model the following example demonstrates that the principal might be better off not using all free labor. It shows that for some OR technology with two agents, for some values the principal is better off removing one agent (discarding his free labor) and contracting with the other.

Example 1. Consider an anonymous OR technology with two agents ($n = 2$), $c = 1$ and $\gamma = 1 - \delta = 0.2$. The optimal contract is obtained when the principal contracts with no agent for $0 \leq v \leq 3.65...$, with one agent for $3.65... \leq v \leq 118.75$, and with both agents for $v \geq 118.75$. However, if we allow the principal to ex-ante remove agents from the network, then, for example, when $v = 4$, the principal obtains a utility of more than 1.867 if the other agent does not participate, compared to a utility of 1.61, if the other agent does participate. It turns out that for $3.04... \leq v \leq 118.75$, the optimal contract is achieved when the principal contracts with a single agent and removes the second one.

Obviously, this example strikes us as counter-intuitive because there is unutilized "free-labor" – the principal prefers that the second agent will not participate despite the fact that he increases the probability of success with no additional cost. Yet, free labor increases *free riding* which results with a lower utility for the principal overall.

We note that the phenomena of costly free labor has also been identified in work on selfish routing [5,7] and in hiring teams with no hidden-actions [4].

In what follows, we will formally define the concept of free-labor and study technologies in which free labor is always used and technologies in which it does not.

4 The Price of Free-Labor (POFL)

We next like to define a measure of the loss to the principal due to not be able to discard free labor. We begin by formally defining the meaning of wasting free labor.

Recall that our focus here is on motivating agents, rather than on the coordination between agents, thus, we are only interested in (weakly) monotone success functions. That is:

$$\forall i \in N, \forall a_{-i} \in A_{-i} \quad a_i \succ_i a'_i \Rightarrow t(a_i, a_{-i}) \geq t(a'_i, a_{-i})$$

Definition 3. *For a given agent i, action $a_i \in A_i$* wastes free-labor *if there exists an action $a'_i \in A_i$, such that $a'_i \succ_i a_i$ while $c(a'_i) = c(a_i)$.*

Note that if a_i wastes free labor then it is possible to (weakly) improve the project success by moving to a'_i with no increase in cost. The *contract $a \in A$ wastes free labor* if for some agent i, action a_i wastes free-labor. The two action profiles $a' \in A$ and $a \in A$ *correspond to the same costs* if for any agent i, $c(a'_i) = c(a_i)$.

Definition 4. *Given a technology $(t, c(\cdot))$ with agents' action spaces $A_1, \ldots, A_n$, the sub-technology that* utilizes all free-labor *is the technology $(t, c(\cdot))$ with agents' action spaces $A'_1, \ldots, A'_n$, obtained by restricting the action space for each agent i to the set of actions that does not waste free labor, that is*
$A'_i = \{a_i \in A_i | a_i$ does not waste free-labor$\}$.

The sub-technology that utilizes all free-labor restricts each agent to actions that do no waste free-labor. In the particular case of structured technologies with the "removal" model, this means that no agent is ever removed.

We are now ready to define the measure on the damage to the principal if he is restricted to the sub-technology that utilizes all free-labor.

Definition 5. *The* price of free-labor *$POFL(t, c(\cdot))$ of a technology $(t, c(\cdot))$ is defined as the ratio between the principal's utility under the optimal contract, and the principal's utility under the optimal contract in the case that he is restricted to the sub-technology that utilizes all free-labor.*

Formally, for a given value v, let $a^*(v) \in A_1 \times \ldots \times A_n = A$ be an optimal contract for v in A, and let $e^*(v) \in A'_1 \times \ldots \times A'_n = A'$ be an optimal contract for v in the sub-technology that utilizes all free-labor (with action spaces A' as defined in Definition 4). The price of free-labor is defined to be

$$POFL(t, c(\cdot)) = Sup_{v>0} \frac{u(a^*(v), v)}{u(e^*(v), v)}$$

By definition we need to find the supremum over a continuum of values. Yet, we are able to show that the POFL is obtained at one of finitely many important points, the transition points between optimal contracts.

Lemma 2. *For any technology $(t, c(\cdot))$ with finite action spaces ($|A_i| < \infty$ for all $i \in N$) the price of free-labor is obtained at a transition point (of either the original technology or the sub-technology with no waste of free-labor).*

Note that the lemma implies that the POFL is obtained, and that it is obtained at a finite positive value.

5 Technologies with Trivial POFL

In this section we consider general technologies and identify a set of technologies for which the POFL is 1, and no free-labor is ever wasted. We need one additional technical condition. A cost function $c_i : A_i \rightarrow \Re_+$ has *finite image* if there exists a number $K < \infty$ such that $|Image(c_i)| < K$. This means that there are only finitely different possible costs for all the actions[7]. A technology $(t, c(\cdot))$ has *finite cost image* if for any $i \in N$, the cost function $c_i(\cdot)$ has a finite image.

Theorem 1. *For any technology $(t, c(\cdot))$ that exhibits IRS and has finite image, the price of free-labor is 1. That is, for any value v, there exists an optimal contract (out of A) that does not waste any free labor.*

The theorem presents a family of technologies for which the price of free-labor is trivial. A natural question is at what extend this family is maximal. In the next section we show that for structured technologies it is maximal in a sense. Specifically, we show that for any function that is not AND (which ensures IRS), there are parameters such that the price is not trivial.

6 Sub-technologies: Only *AND* Ensures Trivial POFL

In the previous section we have seen that technologies that exhibit IRS have trivial POFL. It is easy to show that AND technology exhibits IRS (even in the "removal" model).

Observation 2. *The AND technology exhibits IRS.*

From Theorem 1 we derive the following corollary.

Corollary 1. *The price of free-labor for AND technology in the "removal" model is trivial (1).*

For the "removal" model we can actually present a weaker condition than IRS that ensures that there exists an optimal contract that is non-excluding (all agents participate, none removed). The new condition requires that for any agent i, the increase in success probability when he changes his action from shirking to exerting effort, (weakly) increases when all removed agents are added (becoming participating agents). This condition (which is formally defined and discussed at the full version of the paper) is sufficient to ensure the existence of an optimal contract that is non-excluding. Which structured technologies satisfy this condition? A technology is determined by the Boolean success function and the parameters of the agents. We are interested in finding with functions ensures that the technology has trivial POFL for any choice of agents' parameters.

We show that the AND function is the *only* monotone function which *ensures* that $POFL$ is trivial, out of all technologies that are based on a Boolean function. That is, given any monotone Boolean function that is not an AND function, there exist values for γ_i and δ_i such that the POFL is greater than 1. This is a result of the fact that any non-AND function has an OR function "embedded" in it, and for OR, by Example 1, there exists a constant $\zeta > 1$ such that $POFL > \zeta$.

[7] The actions space A_i may still be infinite.

Lemma 3. *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ for $n \geq 2$ be a monotone Boolean function that is not constant and not a conjunction of some subset of the input bits. Then there exists an assignment to all but two of the bits such that the restricted function is a disjunction of the two bits.*

Finally we present the main result of this section, showing that the AND function is the only function that ensures trivial POFL.

Corollary 2. *Let f be any monotone Boolean function that is not constant and not a conjunction of some subset of the input bits (an AND function). Then there exists a set of parameters $\{\gamma_i, \delta_i\}_{i \in N}$ such that the POFL of the structured technology with the above parameters (and identical cost $c = 1$) is greater than ζ, for some constant $\zeta > 1$.*

7 Process-Based vs. Function-Based Technologies

We now present another natural example that may be viewed as having implications on the controversy of *process-based* (PB) versus *function-based* (FB) team formation approaches [10,14]. In the PB approach, each member of the team is in charge of a different stage in the production process of a single product, and the product is successfully produced only if all stages have succeeded in at least one team. In contrast, an FB team accommodates agents who all work on the same stage of the production process, and the product is successfully produced if there was at least one successful agent in each stage.

The PB and FB approaches can be represented by the OOA and AOO networks, respectively. Clearly, in the FB approach the product will be produced with higher probability (since in the PB approach, a failure of a single stage determines a failure of his team's product). However, in the hidden-actions case the principal sometimes favor PB teams due to the high level of free-riding in FB teams, as demonstrated in the following example.

Example 2. Consider the network demonstrated in Figure 3, where the middle edge connects the middle points of the upper and lower paths, and has a success probability of η. Both the OAA and the AOO networks with $n_l = 2$ and $n_c = 2$ are special cases of this network with $\eta = 0$ and $\eta = 1$, respectively.

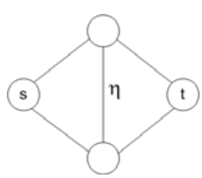


Fig. 3. A network that exhibits Braess-like paradox. As μ changes from 0 to 1, the network moves from Process-Based to Function-Based (and from OOA to AOO).

Clearly, the probability that a message sent from node s reaches node t is better when $\eta = 1$; namely, in the AOO network. This implies that for sufficiently large value of v, AOO is better for the principal. Nevertheless, due to the high level of free-riding in the AOO network compared to OOA, there exist values for which the optimal contract under the OOA network achieves a better utility than the AOO network. For example, in the case that for all i, $\gamma_i = 1 - \delta_i = 0.2$, $c_i = 1$, and $v = 110$, the optimal contract in the AOO network is to contract with one agent from each OR-component, which yields utility of 74.17..., while in the OOA network, the optimal utility level is 75.59.., which is achieved when contracting with all four agents.

One can think of the edge that succeeds with probability η as an edge that is controlled by an agent with cost of 0 to supply both $\eta = 0$ and $\eta = 1$. Our example above can be viewed as showing that the principal is better off wasting the free labor of that agent as for the presented paraments he prefers that agent to take the action with $\eta = 0$ (although the agent can supply $\eta = 1$ with no additional cost) as it decreases free riding by the other agents.

References

1. Babaioff, M., Feldman, M., Nisan, N.: Combinatorial Agency. In: ACM Conference on Electronic Commerce (EC 2006), pp. 18–28 (2006)
2. Babaioff, M., Feldman, M., Nisan, N.: Mixed strategies in combinatorial agency. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 353–364. Springer, Heidelberg (2006)
3. Che, Y.K., Yoo, S.W.: Optimal Incentives in Teams. American Economic Review 91, 525–541 (2001)
4. Chen, N., Karlin, A.R.: Cheap labor can be expensive. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), pp. 707–715 (2007)
5. Cole, R., Dodis, Y., Roughgarten, T.: Pricing Networks with Selfish Routing. In: Workshop on Economics of Peer-to-Peer Systems (2003)
6. Eidenbenz, R., Schmid, S.: Combinatorial agency with audits. In: IEEE International Conference on Game Theory for Networks, GameNets (2009)
7. Elkind, E.: The Costs of Cheap Labor Are Hard to Measure: Edge Deletion and VCG Payments in Graphs. In: ACM Conference on Electronic Commerce (2005)
8. Holmstrom, B.: Moral Hazard in Teams. Bell Journal of Economics 13, 324–340 (1982)
9. Itoh, H.: Incentives to Help Multi-Agent Situations. Econometrica 59, 611–636 (1991)
10. Malone, T.W., Smith, S.A.: Modeling the Performance of Organizational Structures. Operations Research 36, 421–436 (1988)
11. Mass-Colell, A., Whinston, M., Green, J.: Microeconomic Theory. Oxford University Press, Oxford (1995)
12. Rotemberg, J.J.: Process- Versus Function-Based Hierarchies. Journal of Economics and Management Strategy 8, 453–487 (1999)
13. Strausz, R.: Moral Hazard in Sequential Teams. Departmental Working Paper. Free University of Berlin (1996)
14. Winter, E.: Collocation and Incentives: Function vs. Process-Based Teams. In: Working Papaer (2005)

The Cost of Stability in Coalitional Games

Yoram Bachrach[1], Edith Elkind[2,3], Reshef Meir[4], Dmitrii Pasechnik[3], Michael Zuckerman[4], Jörg Rothe[5], and Jeffrey S. Rosenschein[4]

[1] Microsoft Research, Cambridge, United Kingdom
[2] University of Southampton, United Kingdom
[3] Nanyang Technological University, Singapore
[4] School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
[5] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Germany

Abstract. A key question in cooperative game theory is that of coalitional stability, usually captured by the notion of the *core*—the set of outcomes such that no subgroup of players has an incentive to deviate. However, some coalitional games have empty cores, and any outcome in such a game is unstable.

In this paper, we investigate the possibility of stabilizing a coalitional game by using external payments. We consider a scenario where an external party, which is interested in having the players work together, offers a supplemental payment to the grand coalition (or, more generally, a particular coalition structure). This payment is conditional on players not deviating from their coalition(s). The sum of this payment plus the actual gains of the coalition(s) may then be divided among the agents so as to promote stability. We define the *cost of stability (CoS)* as the minimal external payment that stabilizes the game.

We provide general bounds on the cost of stability in several classes of games, and explore its algorithmic properties. To develop a better intuition for the concepts we introduce, we provide a detailed algorithmic study of the cost of stability in weighted voting games, a simple but expressive class of games which can model decision-making in political bodies, and cooperation in multiagent settings. Finally, we extend our model and results to games with coalition structures.

1 Introduction

In recent years, algorithmic game theory, an emerging field that combines computer science, game theory and social choice, has received much attention from the multiagent community [19,8,22,20]. Indeed, multiagent systems research focuses on designing intelligent agents, i.e., entities that can coordinate, cooperate and negotiate without requiring human intervention. In many application domains, such agents are *self-interested*, i.e., they are built to maximize the rewards obtained by their creators. Therefore, these agents can be modeled naturally using game-theoretic tools. Moreover, as agents often have to function in rapidly changing environments, computational considerations are of great concern to their designers as well.

In many settings, such as online auctions and other types of markets, agents act individually. In this case, the standard notions of noncooperative game theory, such as *Nash equilibrium* or *dominant-strategy equilibrium*, provide a prediction of the outcome of the interaction. However, another frequently occurring type of scenario is that agents

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 122–134, 2009.

need to form teams to achieve their individual goals. In such domains, the focus turns from the interaction between single agents to the capabilities of subsets, or *coalitions*, of agents. Thus, a more appropriate modeling toolkit for this setting is that of *cooperative*, or *coalitional*, game theory [4], which studies what coalitions are most likely to arise, and how their members distribute the gains from cooperation. When agents are self-interested, the latter question is obviously of great importance. Indeed, the *total* utility generated by the coalition is of little interest to individual agents; rather, each agent aims to maximize her own utility. Thus, a *stable* coalition can be formed only if the gains from cooperation can be distributed in a way that satisfies all agents.

The most prominent solution concept that aims to formalize the idea of stability in coalitional games is the *core*. Informally, an *outcome* of a coalitional game is a *payoff vector* which for each agent lists her share of the profit of the *grand coalition*, i.e., the coalition that includes all agents. An outcome is said to be in the core if it distributes gains so that no subset of agents has an incentive to abandon the grand coalition and form a coalition of their own. It can be argued that the concept of the core captures the intuitive notion of stability in cooperative settings. However, it has an important drawback: the core of a game may be empty. In games with empty cores, any outcome is unstable, and therefore there is always a group of agents that is tempted to abandon the existing plan. This observation has triggered the invention of less demanding solution concepts, such as ε-core and the least core, as well as an interest in noncooperative approaches to identifying stable outcomes in coalitional games [5,17].

In this paper, we approach this issue from a different perspective. Specifically, we examine the possibility of stabilizing the outcome of a game using external payments. Under this model, an external party (the *center*), which can be seen as a central authority interested in stable functioning of the system, attempts to incentivize a coalition of agents to cooperate in a stable manner. This party does this by offering the members of a coalition a supplemental payment if they cooperate. This external payment is given to the coalition as a whole, and is provided only if this coalition is formed.

Clearly, when the supplemental payment is large enough, the resulting outcome is stable: the profit that the deviators can make on their own is dwarfed by the subsidy they could receive by sticking to the prescribed solution. However, normally the external party would want to minimize its expenditure. Thus, in this paper we define and study the *cost of stability*, which is the minimal supplemental payment that is required to ensure stability in a coalitional game. We start by considering this concept in the context where the central authority aims to ensure that *all* agents cooperate, i.e., it offers a supplemental payment in order to stabilize the grand coalition. We then extend our analysis to the setting where the goal of the center is the stability of a *coalition structure*, i.e., a partition of all agents into disjoint coalitions. In this setting, the center does not expect the agents to work as a single team, but nevertheless wants each individual team to be immune to deviations. Finally, we consider the scenario where the center is concerned with the stability of a particular coalition within a coalition structure. This model is appropriate when the central authority wants a particular group of agents to work together, but is indifferent to other agents switching coalitions.

We first provide bounds on the cost of stability in general coalitional games. We then show that for some interesting special cases, such as super-additive games, these bounds can be improved considerably. We also propose a general algorithmic technique for computing the cost of stability. Then, to develop a better understanding of the concepts proposed in the paper, we apply them in the context of *weighted voting games* (WVGs), a simple but powerful class of games that have been used to model cooperation in settings as diverse as, on the one hand, decision-making in political bodies such as the United Nations Security Council and the International Monetary Fund and, on the other hand, resource allocation in multiagent systems. For such games, we are able to obtain a complete characterization of the cost of stability from an algorithmic perspective.

The paper is organized as follows. In Section 2, we provide the necessary background on coalitional games. In Section 3, we formally define the cost of stability for the setting where the desired outcome is the grand coalition, prove bounds on the cost of stability, and outline a general technique for computing it. We then focus on the computational aspects of the cost of stability in the context of our selected domain, i.e., weighted voting games. In Section 4.1, we demonstrate that computing the cost of stability in such games is coNP-hard if the weights are given in binary. On the other hand, for unary weights, we provide an efficient algorithm for this problem. We also investigate whether the cost of stability can be efficiently approximated. In Section 4.2, we answer this question positively by describing a fully polynomial-time approximation scheme (FPTAS) for our problem. We complement this result by showing that, by distributing the payments in a very natural manner, we get within a factor of 2 of the optimal adjusted gains, i.e., the sum of the value of the grand coalition and the external payments. While this method of allocating payoffs does not necessarily minimize the center's expenditure, the fact that it is both easy to implement and has a bounded worst-case performance may make it an attractive proposition in certain settings. In Section 5, we extend our discussion to the setting where the center aims to stabilize an arbitrary coalition structure, or a particular coalition within it, rather than the grand coalition. We end the paper with a discussion of related work and some conclusions.

We omit some of the proofs due to space constraints; the full version of the paper (with all proofs included) is available online [2]. A preliminary version of this paper was published in AAMAS'09 [3].

2 Preliminaries

Throughout this paper, given a vector $\boldsymbol{x} = (x_1, \dots, x_n)$ and a set $C \subseteq \{1, \dots, n\}$ we write $x(C)$ to denote $\sum_{i \in C} x_i$.

Definition 1. *A* (transferable utility) coalitional game $G = (I, v)$ *is given by a set of* agents *(synonymously,* players*)* $I = \{1, \dots, n\}$ *and a* characteristic function $v : 2^I \to \mathbb{R}^+ \cup \{0\}$ *that for any subset (coalition) of agents lists the total utility these agents achieve by working together. We assume* $v(\emptyset) = 0$.

A coalitional game $G = (I, v)$ is called *increasing* if for all coalitions $C' \subseteq C$ we have $v(C') \leq v(C)$, and *super-additive* if for all disjoint coalitions $C, C' \subseteq I$ we have $v(C) + v(C') \leq v(C \cup C')$. Note that since $v(C) \geq 0$ for any $C \subseteq I$, all super-additive games are increasing. A coalitional game $G = (I, v)$ is called *simple* if it is increasing and $v(C) \in \{0, 1\}$ for all $C \subseteq I$. In a simple game, we say that a coalition $C \subseteq I$ *wins* if $v(C) = 1$, and *loses* if $v(C) = 0$. Finally, a coalitional game is called *anonymous* if $v(C) = v(C')$ for any $C, C' \subseteq I$ such that $|C| = |C'|$. A particular class of simple games considered in this paper is that of *weighted voting games* (WVGs).

Definition 2. *A* weighted voting game *is a simple coalitional game given by a set of agents* $I = \{1, \ldots, n\}$*, a vector* $\mathbf{w} = (w_1, \ldots, w_n)$ *of nonnegative weights, where* w_i *is agent* i*'s weight, and a threshold* q*. The* weight of a coalition $C \subseteq I$ *is* $w(C) = \sum_{i \in C} w_i$*. A coalition* C wins the game *(i.e.,* $v(C) = 1$*) if* $w(C) \geq q$*, and* loses the game *(i.e.,* $v(C) = 0$*) if* $w(C) < q$*.*

We denote the WVG with the weights $\mathbf{w} = (w_1, \ldots, w_n)$ and the threshold q as $[\mathbf{w}; q]$ or $[w_1, \ldots, w_n; q]$. Also, we set $w_{\max} = \max_{i \in I} w_i$. It is easy to see that WVGs are simple games; however, they are not necessarily super-additive. Throughout this paper, we assume that $w(I) \geq q$, i.e., the grand coalition wins.

The characteristic function of a coalitional game defines only the *total* gains a coalition achieves, but does not offer a way of distributing them among the agents. Such a division is called an imputation (or, sometimes, a payoff vector).

Definition 3. *Given a coalitional game* $G = (I, v)$*, a vector* $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$ *is called an* imputation for G *if it satisfies* $p_i \geq v(\{i\})$ *for each* i*,* $1 \leq i \leq n$*, and* $\sum_{i=1}^{n} p_i = v(I)$*. We call* p_i *the* payoff of agent i*; the* total payoff of a coalition $C \subseteq I$ *is given by* $p(C)$*. We write* $\mathcal{I}(G)$ *to denote the set of all imputations for* G*.*

For an imputation to be stable, it should be the case that no subset of players has an incentive to deviate. Formally, we say that a coalition C *blocks* an imputation $\mathbf{p} = (p_1, \ldots, p_n)$ if $p(C) < v(C)$. The *core* of a coalitional game G is defined as the set of imputations not blocked by any coalition, i.e., $\mathrm{core}(G) = \{\mathbf{p} \in \mathcal{I}(G) \mid p(C) \geq v(C) \text{ for each } C \subseteq I\}$. An imputation in the core guarantees the stability of the grand coalition. However, the core can be empty.

In WVGs, and, more generally, in simple games, one can characterize the core using the notion of veto agents, i.e., agents that are indispensable for forming a winning coalition. Formally, given a simple coalitional game $G = (I, v)$, an agent $i \in I$ is said to be a *veto agent* if for all coalitions $C \subseteq I \setminus \{i\}$ we have $v(C) = 0$. The following is a folklore result regarding nonemptiness of the core.

Theorem 1. *Let* $G = (I, v)$ *be a simple coalitional game. If there are no veto agents in* G*, then the core of* G *is empty. Otherwise, let* $I' = \{i_1, \ldots, i_m\}$ *be the set of veto agents in* G*. Then the core of* G *is the set of imputations that distribute all the gains among the veto agents only, i.e.,* $\mathrm{core}(G) = \{\mathbf{p} \in \mathcal{I}(G) \mid p(I') = 1\}$.

So far, we have tacitly assumed that the only possible outcome of a coalitional game is the formation of the grand coalition. However, often it makes more sense for the agents to form several disjoint coalitions, each of which can focus on its own task. For example, WVGs can be used to model the setting where each agent has a certain amount of resources (modeled by her weight), and there are a number of identical tasks each of which requires a certain amount of these resources (modeled by the threshold) to be completed. In this setting, the formation of the grand coalition means that only one task will be completed, even if there are enough resources for several tasks.

The situation when agents can split into teams to work on several tasks simultaneously can be modeled using the notion of a coalition structure, i.e., a partition of the set of agents into disjoint coalitions. Formally, we say that $CS = (C^1, \dots, C^m)$ is a *coalition structure* over a set of agents I if $\bigcup_{i=1}^m C^i = I$ and $C^i \cap C^j = \emptyset$ for all $i \neq j$; we write $CS \in \mathcal{CS}(I)$. Also, we overload notation by writing $v(CS)$ to denote $\sum_{C^j \in CS} v(C^j)$. If coalition structures are allowed, an outcome of a game is not just an imputation, but a pair $(CS, \mathbf{p})$, where $\mathbf{p}$ is an imputation for the coalition structure CS, i.e., $\mathbf{p}$ distributes the gains of every coalition in CS among its members. Formally, we say that $\mathbf{p} = (p_1, \dots, p_n)$ is an *imputation for a coalition structure* $CS = (C^1, \dots, C^m)$ in a game $G = (I, v)$ if $p_i \geq 0$ for all i, $1 \leq i \leq n$, and $p(C^j) = v(C^j)$ for all j, $1 \leq j \leq m$; we write $\mathbf{p} \in \mathcal{I}(CS, G)$. We can also generalize the notion of the core introduced earlier in this section to games with coalition structures. Namely, given a game $G = (I, v)$, we say that an outcome $(CS, \mathbf{p})$ is in the *CS-core of* G if CS is a coalition structure over I, $\mathbf{p} \in \mathcal{I}(CS, G)$ and $p(C) \geq v(C)$ for all $C \subseteq I$; we write $(CS, \mathbf{p}) \in$ *CS-core*(G). Note that if $\mathbf{p}$ is in the core of G then $(I, \mathbf{p})$ is in the CS-core of G; however, the converse is not necessarily true.

3 The Cost of Stability

In many games, forming the grand coalition maximizes social welfare; this happens, for example, in super-additive games. However, the core of such games may still be empty. In this case, it would be impossible to distribute the gains of the grand coalition in a stable way, so it may fall apart despite being socially optimal. Thus, an external party, such as a benevolent central authority, may want to incentivize the agents to cooperate, e.g., by offering the agents a supplemental payment Δ if they stay in the grand coalition. This situation can be modeled as an *adjusted coalitional game* derived from the original coalitional game G.

Definition 4. *Given a coalitional game* $G = (I, v)$ *and* $\Delta \geq 0$*, the* adjusted coalitional game $G(\Delta) = (I, v')$ *is given by* $v'(C) = v(C)$ *for* $C \neq I$*, and* $v'(I) = v(I) + \Delta$*.*

We call $v'(I) = v(I) + \Delta$ the *adjusted gains* of the grand coalition. We say that a vector $\mathbf{p} \in \mathbb{R}^n$ is a *super-imputation* for a game $G = (I, v)$ if $p_i \geq 0$ for all $i \in I$ and $p(I) \geq v(I)$. Furthermore, we say that a super-imputation $\mathbf{p}$ is *stable* if $p(C) \geq v(C)$ for all $C \subseteq I$. A super-imputation $\mathbf{p}$ with $p(I) = v(I) + \Delta$ distributes the adjusted gains, i.e., it is an imputation for $G(\Delta)$; it is stable if and only if it is in the core of $G(\Delta)$. We say that a supplemental payment Δ *stabilizes* the grand coalition in a game G if the adjusted game $G(\Delta)$ has a nonempty core. Clearly, if Δ is large enough (e.g., $\Delta = n \max_{C \subseteq I} v(C)$), the game $G(\Delta)$ will have a nonempty core. However, usually the central authority wants to spend as little money as possible. Hence, we define the cost of stability as the *smallest* external payment that stabilizes the grand coalition.

Definition 5. *Given a coalitional game* $G = (I, v)$*, its* cost of stability $CoS(G)$ *is defined as* $CoS(G) = \inf\{\Delta \mid \Delta \geq 0 \text{ and } \mathrm{core}(G(\Delta)) \neq \emptyset\}$.

We have argued that the set $\{\Delta \mid \Delta \geq 0 \text{ and } \mathrm{core}(G(\Delta)) \neq \emptyset\}$ is nonempty. Therefore, $G(\Delta)$ is well-defined. Now, we prove that this set contains its greatest lower bound

$CoS(G)$, i.e., that the game $G(CoS(G))$ has a nonempty core. While this can be shown using a continuity argument, we will now give a different proof, which will also be useful for exploring the cost of stability from an algorithmic perspective. Fix a coalitional game $G = (I, v)$ and consider the following linear program $\mathcal{LP}^*$:

$$\min \Delta \quad \text{subject to:}$$

$$\Delta \geq 0, \tag{1}$$

$$p_i \geq 0 \quad \text{for each } i = 1, \ldots, n, \tag{2}$$

$$\sum_{i \in I} p_i = v(I) + \Delta, \tag{3}$$

$$\sum_{i \in C} p_i \geq v(C) \quad \text{for all } C \subseteq I. \tag{4}$$

It is not hard to see that the optimal value of this linear program is exactly $CoS(G)$. Moreover, any optimal solution of $\mathcal{LP}^*$ corresponds to an imputation in the core of $G(CoS(G))$ and therefore the game $G(CoS(G))$ has a nonempty core.

As an example, consider a uniform weighted voting game, i.e., a WVG $G = [\mathbf{w}; q]$ with $w_1 = \cdots = w_n = w$. We can derive an explicit formula for $CoS(G)$.

Theorem 2. *For a WVG $G = [w, w, \ldots, w; q]$, we have $CoS(G) = \frac{n}{\lceil q/w \rceil} - 1$.*

For example, if $w(n-1) < q \leq wn$, then $CoS(G) = 0$, i.e., G has a nonempty core. On the other hand, if $w = 1$, $n = 3k$ and $q = 2k$ for some integer $k > 0$, i.e., $q = \frac{2}{3}n$, we have $CoS(G) = \frac{3}{2} - 1 = \frac{1}{2}$.

3.1 Bounds on $CoS(G)$ in General Coalitional Games

Consider an arbitrary coalitional game $G = (I, v)$. Clearly, $CoS(G) = 0$ if and only if G has a nonempty core. Further, we have argued that $CoS(G)$ is upper-bounded by $n \max_{C \subseteq I} v(C)$, i.e., $CoS(G)$ is finite for any fixed coalitional game. Moreover, the bound of $n \max_{C \subseteq I} v(C)$ is (almost) tight. To see this, consider a (simple) game G' given by $v'(\emptyset) = 0$ and $v'(C) = 1$ for all $C \neq \emptyset$. Clearly, we have $CoS(G') = n - 1$: any super-imputation that pays some agent less than 1 will not be stable, whereas setting $p_i = 1$ for all $i \in I$ ensures stability. Thus, the cost of stability can be quite large relative to the value of the grand coalition.

On the other hand, we can provide a lower bound on $CoS(G)$ in terms of the values of coalition structures over I. Indeed, for an arbitrary coalition structure $CS \in \mathcal{CS}(I)$, we have $CoS(G) \geq v(CS) - v(I)$. To see this, note that if the total payment to the grand coalition is less than $(v(CS) - v(I)) + v(I)$, then for some coalition $C \in CS$ it will be the case that $p(C) < v(C)$. It would be tempting to conjecture that $CoS(G) = \max_{CS \in \mathcal{CS}(I)}(v(CS) - v(I))$. However, a counterexample is provided by Theorem 2 with $w = 1$, $q = \frac{2}{3}n$: indeed, in this case we have $CoS(G) = \frac{1}{2}$, yet $\max_{CS \in \mathcal{CS}(I)}(v(CS) - v(I)) = 0$. We can summarize these observations as follows.

Theorem 3. *For any coalitional game $G = (I, v)$, we have*

$$\max_{CS \in \mathcal{CS}(I)}(v(CS) - v(I)) \leq CoS(G) \leq n \max_{C \subseteq I} v(C).$$

For super-additive games, we can strengthen the upper bound considerably. Note that in such games the grand coalition maximizes social welfare, so its stability is particularly desirable. Yet, as the second part of Theorem 4 implies, ensuring stability may turn out to be quite costly even in this restricted setting.

Theorem 4. *For any super-additive game $G = (I, v)$, $|I| = n$, we have $CoS(G) \leq (\sqrt{n} - 1)v(I)$, and this bound is asymptotically tight.*

For anonymous super-additive games, further improvements are possible.

Theorem 5. *For any anonymous super-additive game $G = (I, v)$, we have $CoS(G) \leq 2v(I)$, and this bound is asymptotically tight.*

A somewhat similar stability-related concept is the *least core*, which is the set of all imputations $\mathbf{p}$ that minimize the maximal *deficit* $v(C) - p(C)$. In particular, the *value* of the least core $\varepsilon(G)$, defined as $\varepsilon(G) = \inf_{\mathbf{p}\in\mathcal{I}(G)}\{\max\{v(C) - p(C) \mid C \subseteq I\}\}$, is strictly positive if and only if the cost of stability is strictly positive. The following proposition provides a more precise description of the relationship between the value of the least core and the cost of stability.

Proposition 1. *For any coalitional game $G = (I, v)$ with $|I| = n$ such that $\varepsilon(G) \geq 0$, we have $CoS(G) \leq n\varepsilon(G)$, and this bound is asymptotically tight.*

3.2 Algorithmic Properties of $CoS(G)$

The linear program $\mathcal{LP}^*$ provides a way of computing $CoS(G)$ for any coalitional game G. However, this linear program contains exponentially many constraints (one for each subset of I). Thus, solving it directly would be too time-consuming for most games. Note that for general coalitional games, this is, in a sense, inevitable: in general, a coalitional game is described by its characteristic function, i.e., a list of 2^n numbers. Thus, to discuss the algorithmic properties of $CoS(G)$, we need to restrict our attention to games with compactly representable characteristic functions.

A standard approach to this issue is to consider games that can be described by polynomial-size circuits. Formally, we say that a class $\mathcal{G}$ of games has a *compact circuit representation* if there exists a polynomial p such that for every $G \in \mathcal{G}$, $G = (I, v)$, $|I| = n$, there exists a circuit $\mathcal{C}$ of size $p(n)$ with n binary inputs that on input $(b_1, \ldots, b_n)$ outputs $v(C)$, where $C = \{i \in I \mid b_i = 1\}$.

Unfortunately, it turns out that having a compact circuit representation does not guarantee efficient computability of $CoS(G)$. Indeed, it is easy to see that WVGs with integer weights have such a representation. However, in the next section we will show that computing $CoS(G)$ for such games is computationally intractable (Theorem 7). We can, however, provide a *sufficient* condition for $CoS(G)$ to be efficiently computable. To do so, we will first formally state the relevant computational problems.

SUPER-IMPUTATION-STABILITY: Given a coalitional game G (compactly represented by a circuit), a supplemental payment Δ and an imputation $\mathbf{p} = (p_1, \ldots, p_n)$ in the adjusted game $G(\Delta)$, decide whether $\mathbf{p} \in \mathrm{core}(G(\Delta))$.

COS: Given a coalitional game G (compactly represented by a circuit) and a parameter Δ, decide whether $CoS(G) \leq \Delta$, i.e., whether $\mathrm{core}(G(\Delta)) \neq \emptyset$.

Consider first SUPER-IMPUTATION-STABILITY. Fix a game $G = (I, v)$. For any super-imputation $\mathbf{p}$ for G, let $d(G, \mathbf{p}) = \max_{C \subseteq I}(v(C) - p(C))$ be the maximum deficit of a coalition under $\mathbf{p}$. Clearly, $\mathbf{p}$ is stable if and only if $d(G, \mathbf{p}) \leq 0$. Observe also that for any $\Delta > 0$ it is easy to decide whether $\mathbf{p}$ is an imputation for $G(\Delta)$. Thus, a polynomial-time algorithm for computing $d(G, \mathbf{p})$ can be converted into a polynomial-time algorithm for SUPER-IMPUTATION-STABILITY. Further, we can decide COS via solving $\mathcal{LP}^*$ by the ellipsoid method. The ellipsoid method runs in polynomial time given a polynomial-time *separation oracle*, i.e., a procedure that takes as input a candidate feasible solution, checks if it indeed is feasible, and if this is not the case, returns a violated constraint. Now, given a vector $\mathbf{p}$ and a parameter Δ, we can easily check if they satisfy constraints (1)–(3), i.e., if $\mathbf{p}$ is an imputation for $G(\Delta)$. To verify constraint (4), we need to check if $\mathbf{p}$ is in the core of $G(\Delta)$. As argued above, this can be done by checking whether $d(G, \mathbf{p}) \leq 0$. We summarize these results as follows.

Theorem 6. *Consider a class of coalitional games $\mathcal{G}$ with a compact circuit representation. If there is an algorithm that for any $G \in \mathcal{G}$, $G = (I, v)$, $|I| = n$, and for any super-imputation $\mathbf{p}$ for G computes $d(G, \mathbf{p})$ in time $\mathrm{poly}(n, |\mathbf{p}|)$, where $|\mathbf{p}|$ is the number of bits in the binary representation of $\mathbf{p}$, then for any $G \in \mathcal{G}$ the problems* SUPER-IMPUTATION-STABILITY *and* COS *are polynomial-time solvable.*

We mention in passing that for games with poly-time computable characteristic functions both problems are in coNP. For SUPER-IMPUTATION-STABILITY, the membership is trivial; for COS, it follows from the fact that the game $G(\Delta)$ has a poly-time computable characteristic function as long as G does, and hence we can apply the results of [14] (see the proof of Theorem 7 for details).

4 Cost of Stability in WVGs without Coalition Structures

In this section, we focus on computing the cost of stabilizing the grand coalition in WVGs. We start by considering the complexity of exact algorithms for this problem.

4.1 Exact Algorithms

In what follows, unless specified otherwise, we assume that all weights and the threshold are integers given in binary, whereas all other numeric parameters, such as the supplemental payment Δ and the entries of the payoff vector $\mathbf{p}$, are rationals given in binary. Standard results on linear threshold functions [16] imply that WVGs with integer weights have a compact circuit representation. Thus, we can define the computational problems SUPER-IMPUTATION-STABILITY-WVG and COS-WVG by specializing the problems SUPER-IMPUTATION-STABILITY and COS to WVGs. Both of the resulting problems turn out to be computationally hard.

Theorem 7. *The problems* SUPER-IMPUTATION-STABILITY-WVG *and* COS-WVG *are* coNP*-complete.*

The reductions in the proof of Theorem 7 are from PARTITION. Consequently, our hardness results depend in an essential way on the weights being given in binary. Thus, it is natural to ask what happens if the agents' weights are polynomially bounded (or given in unary). It turns out that in this case the results of Section 3.2 imply that SUPER-IMPUTATION-STABILITY-WVG and COS-WVG are in P, since for WVGs with small weights one can compute $d(G, \mathbf{p})$ in polynomial time.

Theorem 8. SUPER-IMPUTATION-STABILITY-WVG *and* COS-WVG *are in* P *when the agents' weights are polynomially bounded (or given in unary).*

4.2 Approximating the Cost of Stability in Weighted Voting Games

For large weights, the algorithms outlined at the end of the previous section may not be practical. Thus, the center may want to trade off its payment and computation time, i.e., provide a slightly higher supplemental payment for which the corresponding stable super-imputation can be computed efficiently. It turns out that this is indeed possible, i.e., $CoS(G)$ can be efficiently approximated to an arbitrary degree of precision.

Theorem 9. *There exists an algorithm $\mathcal{A}(G, \varepsilon)$ that, given a WVG $G = [\mathbf{w}; q]$ in which the weights of all players are nonnegative integers given in binary and a parameter $\varepsilon > 0$, outputs a value Δ that satisfies $CoS(G) \le \Delta \le (1+\varepsilon)CoS(G)$ and runs in time $\mathrm{poly}(n, \log w_{\max}, 1/\varepsilon)$. That is, there exists a fully polynomial-time approximation scheme (FPTAS) for $CoS(G)$.*

Moreover, one can get a 2-approximation to the adjusted gains simply by paying each agent in proportion to her weight, and this bound can be shown to be tight.

Theorem 10. *For any WVG $G = [\mathbf{w}; q]$ with $CoS(G) = \Delta$, the super-imputation $\mathbf{p}^*$ given by $p_i^* = \min\{1, \frac{w_i}{q}\}$ is stable and satisfies $p^*(I) \le 2p(I)$ for any super-imputation $\mathbf{p} \in \mathrm{core}(G(\Delta))$.*

5 Cost of Stability in Games with Coalition Structures

If a coalitional game is not super-additive, the formation of the grand coalition is not necessarily the most desirable outcome: for example, it may be the case that by splitting into several teams the agents can accomplish more tasks than by working together. In such settings, the central authority may want to stabilize a coalition structure, i.e., a partition of agents into teams. We now generalize the cost of stability to such settings.

5.1 Stabilizing a Fixed Coalition Structure

We first consider the setting where the central authority wants to stabilize a particular coalition structure.

Given a coalitional game $G = (I, v)$, a coalition structure $CS = (C^1, \dots, C^m)$ over I and a vector $\boldsymbol{\Delta} = (\Delta^1, \dots, \Delta^m)$, let $G(\boldsymbol{\Delta})$ be the game with the set of agents I and the characteristic function v' given by $v'(C^i) = v(C^i) + \Delta^i$ for $i = 1, \dots, m$ and

$v'(C) = v(C)$ for any $C \notin \{C^1, \dots, C^m\}$. We say that the game $G(\boldsymbol{\Delta})$ is *stable with respect to* CS if there exists an imputation $\mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta}))$ such that $(CS, \mathbf{p})$ is in the CS-core of $G(\boldsymbol{\Delta})$. Also, we say that an external payment Δ *stabilizes* a coalition structure CS with respect to a game G if there exist $\Delta^1 \geq 0, \dots, \Delta^m \geq 0$ such that $\Delta = \Delta^1 + \dots + \Delta^m$ and the game $G(\boldsymbol{\Delta})$ is stable with respect to CS. We are now ready to define the cost of stability of a coalition structure CS in G.

Definition 6. *Given a coalitional game* $G = (I, v)$ *and a coalition structure* $CS = (C^1, \dots, C^m)$ *over* I*, the* cost of stability $CoS(CS, G)$ *of the coalition structure* CS *in* G *is the smallest external payment needed to stabilize* CS*, i.e.,*

$$CoS(CS, G) = \inf\{\sum_{i=1}^{m} \Delta^i \mid \Delta^i \geq 0 \text{ for } i = 1, \dots, m \quad \text{and}$$
$$\exists \mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta})) \quad \text{s.t.} \quad (CS, \mathbf{p}) \in \textit{CS-core}(G(\boldsymbol{\Delta}))\}.$$

Fix a game $G = (I, v)$ and set $v_{\max} = \max_{C \subseteq I} v(C)$. It is easy to see that for any coalition structure $CS = (C^1, \dots, C^m)$ the game $G(\boldsymbol{\Delta})$, where $\Delta^i = |C^i| v_{\max}$, is stable with respect to CS, and therefore $CoS(CS, G)$ is well-defined and satisfies $CoS(CS, G) \leq n v_{\max}$. Moreover, as in the case of games without coalition structures, the value $CoS(CS, G)$ can be obtained as an optimal solution to a linear program. Indeed, we can simply take the linear program $\mathcal{LP}^*$ and replace the constraint $\sum_{i \in I} p_i = v(I) + \Delta$ with the constraint $\sum_{i \in I} p_i = v(CS) + \Delta$. It is not hard to see that the resulting linear program, which we will denote by $\mathcal{LP}^*_{CS}$, computes $CoS(CS, G)$: in particular, the constraints $\Delta^i \geq 0$ for $i = 1, \dots, m$ are implicitly captured by the constraints $\sum_{i \in C^i} p_i \geq v(C^i)$ in line (4) of $\mathcal{LP}^*_{CS}$.

We now turn to the question of computing the cost of stability of a given coalition structure in WVGs. To this end, we will modify the decision problems stated in Section 4.1 as follows.

SUPER-IMPUTATION-STABILITY-WVG-CS: Given a WVG $G = [\mathbf{w}; q]$ with the set of agents I, a coalition structure $CS = (C^1, \dots, C^m)$ over I, a vector $\boldsymbol{\Delta} = (\Delta^1, \dots, \Delta^m)$ and an imputation $\mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta}))$, decide if $(CS, \mathbf{p})$ is in the CS-core of $G(\boldsymbol{\Delta})$.

COS-WVG-CS: Given a WVG $G = [\mathbf{w}; q]$ with the set of agents I, a coalition structure CS over I and a parameter Δ, decide whether $CoS(CS, G) \leq \Delta$.

The results of Section 4.1 immediately imply that both of these problems are computationally hard even for $m = 1$. Moreover, using the results of [9], we can show that SUPER-IMPUTATION-STABILITY-WVG-CS remains coNP-complete even if $\boldsymbol{\Delta}$ is fixed to be $(0, \dots, 0)$. On the other hand, when weights are integers given in unary, both COS-WVG-CS and SUPER-IMPUTATION-STABILITY-WVG-CS are polynomial-time solvable. Indeed, to solve SUPER-IMPUTATION-STABILITY-WVG-CS, one needs to check if there is a coalition C with $w(C) \geq q$, $p(C) < 1$. This can be done using the dynamic programming algorithm from the proof of Theorem 8. Moreover, to solve COS-WVG-CS, we can simply run the ellipsoid algorithm on the linear program $\mathcal{LP}^*_{CS}$ described earlier in this section, using the algorithm for SUPER-IMPUTATION-STABILITY-WVG-CS as a separation oracle. Thus, we obtain the following result.

Theorem 11. *When all players' weights are integers given in unary, the problems* CoS-WVG-CS *and* Super-Imputation-Stability-WVG-CS *are in* P.

Finally, we adapt the approximation algorithm presented in Section 4.2 to this setting.

Theorem 12. *There exists an FPTAS for* $CoS(CS, G)$ *in WVGs.*

5.2 Finding the Cheapest Coalition Structure to Stabilize

So far, we have focused on the setting where the external party wants to stabilize a particular coalition structure. However, it can also be the case that the central authority simply wants to achieve stability, and does not care which coalition structure arises, as long as it can be made stable using as little money as possible. We will now introduce the notion of *cost of stability for games with coalition structures* to capture this type of setting. Recall that $\mathcal{CS}(I)$ denotes the set of all coalition structures over I.

Definition 7. *Given a coalitional game* $G = (I, v)$*, let the* cost of stability for G with coalition structures*, denoted by* $CoS_{CS}(G)$*, be* $\min\{CoS(CS, G) \mid CS \in \mathcal{CS}(I)\}$.

Clearly, one can compute $CoS_{CS}(G)$ by enumerating all coalition structures over I and picking the one with the smallest value of $CoS(CS, G)$. Alternatively, note that the linear program $\mathcal{LP}^*_{CS}$ depends only on the value of the coalition structure CS. Hence, stabilizing all coalition structures with the same total value has the same cost. Moreover, this implies that the cheapest coalition structure to stabilize is the one that maximizes social welfare. Hence, if we could compute the value of the coalition structure CS^* that maximizes social welfare, we could find $CoS_{CS}(G)$ by solving $\mathcal{LP}^*_{CS^*}$.

For WVGs, paper [9] (see Theorem 2 there) shows that if weights are given in binary, it is NP-hard to decide whether a given game has a nonempty CS-core. As this question is equivalent to asking whether $CoS_{CS}(G) = 0$, the latter problem is NP-hard, too. One might hope that computing $CoS_{CS}(G)$ is easy if the weights of all players are given in unary. However, this does not seem to be the case. Indeed, our algorithms for computing the cost of stability in other settings relied on solving the corresponding linear program. To implement this approach in our scenario, we would need to compute the value of the coalition structure that maximizes social welfare. However, a straightforward reduction from 3-Partition, a classic problem that is known to be NP-hard even for unary weights, shows that the latter problem is NP-hard even if weights are given in unary. While this does not immediately imply that computing $CoS_{CS}(G)$ is hard for small weights, it means that finding the cheapest-to-stabilize outcome is NP-hard even if weights are given in unary.

5.3 Stabilizing a Particular Coalition

We now consider the case where the central authority wants a particular group of agents to work together, but does not care about the stability of the overall game. Thus, it wants to identify a coalition structure containing a particular coalition C and the minimal subsidy to the players that ensures that no set of players that includes members of C wants to deviate. We omit the formal definition of the corresponding cost of stability concept, as well as its algorithmic analysis due to space constraints. However, we would like to mention several subtle points that arise in this context. First, one might think that the optimal way to stabilize a coalition is to offer payments to members of this coalition only. However, this turns out not to be true (see [2]). Second, stabilizing a given coalition may be strictly cheaper than stabilizing *any* of the coalition structures that contain it (see [2]). Thus choosing a good definition of the cost of stability of an individual coalition is a nontrivial issue.

6 Related Work

The complexity of various solution concepts in coalitional games is a well-studied topic [6,13,7,23]. In particular, [10] analyzes some important computational aspects of stability in WVGs, proving a number of results on the complexity of the least core and the nucleolus. The complexity of the CS-core in WVGs is studied in [9]. Paper [15] is similar to ours in spirit. It considers the setting where an external party intervenes in order to achieve a certain outcome using monetary payments. However, [15] deals with the very different domain of *non*cooperative games. There are also similarities between our work and the recent research on bribery in elections [11], where an external party pays voters to change their preferences in order to make a given candidate win. A companion paper [18] studies the cost of stability in network flow games.

7 Conclusion

We have examined the possibility of stabilizing a coalitional game by offering the agents additional payments in order to discourage them from deviating, and defined the cost of stability as the minimal total payment that allows a stable division of the gains. We focused on the computational aspects of this concept for weighted voting games. In the setting where the outcome to be stabilized is the grand coalition, we provided a complete picture of the computational complexity of the related decision problems. We then extended our results to settings where agents can form a coalition structure.

There are several lines of possible future research. First, while the focus of this paper was on weighted voting games, the notion of the cost of stability is defined for any coalitional game. Therefore, a natural research direction is to study the cost of stability in other classes of games. Second, we would like to develop a better understanding of the relationship between the cost of stability of a game, and its least core and nucleolus. Finally, it would be interesting to extend the notion of the cost of stability to games with nontransferable utility and partition function games.

Acknowledgments. We thank the reviewers for many helpful comments. This work was supported in part by DFG grants RO 1202/{11-1, 12-1}, the ESF's EUROCORES program LogICCC, the Alexander von Humboldt Foundation's TransCoop program, EPSRC grant GR/T10664/01, ESRC grant ES/F035845/1, NRF Research Fellowship, Singapore Ministry of Education AcRF Tier 1 Grant, and by ISF grant #898/05.

References

1. Aumann, R.J., Hart, S. (eds.): Handbook of Game Theory, with Economic Applications, vol. 2. North-Holland, Amsterdam (1994)
2. Bachrach, Y., Elkind, E., Meir, R., Pasechnik, D., Zuckerman, M., Rothe, J., Rosenschein, J.S.: The cost of stability in coalitional games. Tech. Rep. arXiv:0907.4385 [cs.GT], ACM Comp. Research Repository (July 2009); Preliminary version: [3]
3. Bachrach, Y., Meir, R., Zuckerman, M., Rothe, J., Rosenschein, J.S.: The cost of stability in weighted voting games (extended abstract). In: Proc. of AAMAS-2009, pp. 1289–1290 (2009)
4. Branzei, R., Dimitrov, D., Tijs, S.: Models in Cooperative Game Theory. Springer, Heidelberg (2008)
5. Chatterjee, K., Dutta, B., Sengupta, K.: A noncooperative theory of coalitional bargaining. Review of Economic Studies 60, 463–477 (1993)
6. Conitzer, V., Sandholm, T.: Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In: Proc. of AAAI-2004, pp. 219–225 (2004)
7. Conitzer, V., Sandholm, T.: Complexity of constructing solutions in the core based on synergies among coalitions. Artificial Intelligence 170(6-7), 607–619 (2006)
8. Ephrati, E., Rosenschein, J.S.: The Clarke Tax as a consensus mechanism among automated agents. In: Proc. of IJCAI-1991, pp. 173–178 (1991)
9. Elkind, E., Chalkiadakis, G., Jennings, N.R.: Coalition structures in weighted voting games. In: Proc. of ECAI-2008, pp. 393–397 (2008)
10. Elkind, E., Goldberg, L.A., Goldberg, P.W., Wooldridge, M.: Computational complexity of weighted threshold games. In: Proc. of AAAI-2007, pp. 718–723 (2007)
11. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: The complexity of bribery in elections. In: Proc. of AAAI-2006, pp. 641–646 (2006); Full version to appear in JAIR
12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
13. Ieong, S., Shoham, Y.: Marginal contribution nets: A compact representation scheme for coalitional games. In: Proc. of EC-2005, pp. 193–292 (2005)
14. Malizia, E., Palopoli, L., Scarcello, F.: Infeasibility certificates and the complexity of the core in coalitional games. In: Proc. of IJCAI-2007, pp. 1402–1407 (2007)
15. Monderer, D., Tennenholtz, M.: K-implementation. Journal of Artificial Intelligence Research 21, 37–62 (2004)
16. Muroga, S.: Threshold Logic and its Applications. John Wiley & Sons, Chichester (1971)
17. Okada, A.: A noncooperative coalitional bargaining game with random proposers. Games and Economic Behavior 16, 97–108 (1996)
18. Resnick, E., Bachrach, Y., Meir, R., Rosenschein, J.S.: The cost of stability in network flow games. In: Královič, R., Niwiński, D. (eds.) MFCS 2009. LNCS, vol. 5734, pp. 636–650. Springer, Heidelberg (2009)
19. Rosenschein, J.S., Genesereth, M.R.: Deals among rational agents. In: Proc. of IJCAI-1985, pp. 91–99 (1985)
20. Sandholm, T., Lesser, V.: Issues in automated negotiation and electronic commerce: Extending the contract net framework. In: Proc. of ICMAS-1995, pp. 328–335 (1995)
21. Taylor, A., Zwicker, W.: Simple Games: Desirability Relations, Trading, Pseudoweightings. Princeton University Press, Princeton (1999)
22. Wellman, M.P.: The economic approach to artificial intelligence. ACM Computing Surveys 27, 360–362 (1995)
23. Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., Iwasaki, A.: Coalitional games in open anonymous environments. In: Proc. of AAAI-2005, pp. 509–514 (2005)

Non-clairvoyant Scheduling Games

Christoph Dürr[1] and Kim Thang Nguyen[2,⋆]

[1] CNRS, LIX UMR 7161, Ecole Polytechnique, France
[2] Department of Computer Science, University of Aarhus, Denmark

Abstract. In a scheduling game, each player owns a job and chooses a machine to execute it. While the social cost is the maximal load over all machines (makespan), the cost (disutility) of each player is the completion time of its own job. In the game, players may follow selfish strategies to optimize their cost and therefore their behaviors do not necessarily lead the game to an equilibrium. Even in the case there is an equilibrium, its makespan might be much larger than the social optimum, and this inefficiency is measured by the price of anarchy – the worst ratio between the makespan of an equilibrium and the optimum. Coordination mechanisms aim to reduce the price of anarchy by designing scheduling policies that specify how jobs assigned to a same machine are to be scheduled. Typically these policies define the schedule according to the processing times as announced by the jobs. One could wonder if there are policies that do not require this knowledge, and still provide a good price of anarchy. This would make the processing times be private information and avoid the problem of truthfulness. In this paper we study these so-called *non-clairvoyant* policies. In particular, we study the RANDOM policy that schedules the jobs in a random order without preemption, and the EQUI policy that schedules the jobs in parallel using time-multiplexing, assigning each job an equal fraction of CPU time.

For these models we study two important questions, the existence of Nash equilibria and the price of anarchy. We show under some restrictions that the game under RANDOM policy is a potential game for two unrelated machines but it is not for three or more; for uniform machines, we prove that the game under this policy always possesses a Nash equilibrium by using a novel potential function with respect to a refinement of best-response dynamic. Moreover, we show that the game under the EQUI policy is a potential game.

Next, we analyze the inefficiency of EQUI policy. Interestingly, the (strong) price of anarchy of EQUI, a non-clairvoyant policy, is asymptotically the same as that of the best *strongly local* policy – policies in which a machine may look at the processing time of jobs assigned to it. The result also indicates that knowledge of jobs' characteristics is not necessarily needed.

1 Introduction

With the development of the Internet, large-scale autonomous systems became more and more important. The systems consist of many independent and selfish

⋆ This work is done while the author was student in Ecole Polytechnique, France.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 135–146, 2009.

agents who compete for the usage of shared resources. Every configuration has some social cost, as well as individual costs for every agent. Due to the lack of coordination, the equilibrium configurations may have high cost compared to the global social optimum and this inefficiency can be captured by the *price of anarchy*. It is defined as the ratio between the the worst case performance of Nash equilibrium and the global optimum. Since the behavior of the agents is influenced by the individual costs, it is natural to come up with mechanisms that both force the existence of Nash equilibria and reduce the price of anarchy. The idea is to try to reflect the social cost in the individual costs, so that selfish agents' behaviors result in a socially desired solution. In particular we are interested in scheduling games, where every player has to choose one machine on which to execute its job. The individual cost of a player is the completion time of its job, and the social cost is the largest completion time over all jobs, the *makespan*. For these games, so called *coordination mechanisms* have been studied by [9]. A *coordination mechanism* is a set of *local policies*, one for every machine, that specify a schedule for the jobs assigned to it, and the schedule can depend only on these jobs. Most prior studied policies depend on the processing times and need the jobs to announce their processing times. The jobs could try to influence the schedule to their advantage by announcing not their correct processing times. There are two ways to deal with this issue. One is to design *truthful coordination mechanisms* where jobs have an incentive to announce their real processing times. Another way is to design mechanisms that do not depend on the processing times at all and this is the subject of this paper: we study coordination mechanisms based on so called *non-clairvoyant policies* that we define in this section.

1.1 Preliminaries

Scheduling. The *machine scheduling problem* is defined as follows: we are given n jobs, m machines and each job needs to be scheduled on exactly one machine. In the most general case machine speeds are unrelated, and for every job $1 \leq i \leq n$ and every machine $1 \leq j \leq m$ we are given an arbitrary processing time p_{ij}, which is the time spend by job i on machine j. A schedule σ is a function mapping each job to some machine. The *load* of a machine j in schedule σ is the total processing time of jobs assigned to this machine, i.e., $\ell_j = \sum_{i:\sigma(i)=j} p_{ij}$. The *makespan* of a schedule is the maximal load over all machines, and is the social cost of a schedule. It is NP-hard to compute the *global optimum* even for identical machines, that is when p_{ij} does not depend on j. We denote by OPT the makespan of the optimal schedule.

Machine Environments. We consider four different machine environments, which all have their own justification. The most general environment concerns unrelated machines as defined above and is denoted $R||C_{\max}$. In the *identical* machine scheduling model, denoted $P||C_{\max}$, every job i comes with a length p_i such that $p_{ij} = p_i$ for every machine j. In the *uniform* machine scheduling model, denoted $Q||C_{\max}$, again every job has *length* p_i and every machine j a speed s_j

such that $p_{ij} = p_i/s_j$. For the *restricted identical* machine model, every job i comes with a length p_i and a set of machines S_i on which it can be scheduled, such that $p_{ij} = p_i$ for $j \in S_i$ and $p_{ij} = \infty$ otherwise. In [6] this model is denoted $PMPM||C_{\max}$, and in [21] it is denoted $B||C_{\max}$.

Nash Equilibria. What we described so far are well known and extensively studied classical scheduling problems. But now consider the situation where each of the n jobs is owned by an independent agent. The agents do not care about the social optimum, their goal is to complete their job as soon as possible. In the paper, we concentrate on *pure strategies* where each agent selects a single machine to process its job. Such a mapping σ is called a *strategy profile.* He is aware of the decisions made by other agents and behaves selfishly. From now on we will abuse notation and identify the agent with his job. The *individual cost* of a job is defined as its completion time. A *Nash equilibrium* is a schedule in which no agent has an incentive to unilaterally switch to another machine. A *strong* Nash equilibrium is a schedule that is resilient to deviations of any coalition, i.e., no group of agents can cooperate and change their strategies in such a way that all players in the group strictly decrease their costs. For some given strategy profile, a *best move* of a job i is a strategy (machine) j such that if job i changes to job j, while all other players stick to their strategy, the cost of i decreases strictly. If there is such a best move, we say that this job is *unhappy*, otherwise it is *happy.* In this setting a Nash equilibrium is a strategy profile where all jobs are happy. The *best-response dynamic* is the process of repeatedly choosing an arbitrary unhappy job and changing it to an an arbitrary best move. A *potential game* [22] is a game that admits a potential function. Consequently, a potential game always possesses a Nash equilibrium. However, the existence of equilibrium does not necessarily mean that the game is potential.

Coordination Mechanism. A *coordination mechanism* is a set of *scheduling policies*, one for each machine, that determines how to schedule jobs assigned to a machine. The idea is to connect the individual cost to the social cost, in such a way that the selfishness of the agents will lead to equilibria that have low social cost. How good is a given coordination mechanism? This is measured by the well-known *price of anarchy* (PoA). It is defined as the ratio between the cost of the worst Nash equilibrium and the optimal cost, which is not an equilibrium in general. We also consider the *strong price of anarchy* (SPoA) which is the extension of the price of anarchy applied to strong Nash equilibria.

Policies. A policy is a rule that specifies how the jobs that are assigned to a machine are to be scheduled. We distinguish between *local, strongly local* and *non-clairvoyant* policies. Let S_j be the set of jobs assigned to machine j. A policy is *local* if the scheduling of jobs on machine j depends only on the parameters of jobs in S_j, i.e., it may look at the processing time p_{ik} of a job $i \in S_j$ on any machine k. A policy is *strongly local* if it looks only at the processing time of jobs in S_j on machine j. We call a policy *non-clairvoyant* if the scheduling of jobs on machine j does not depend on the processing time of any job on any machine.

In this paper we only study coordination mechanisms that use the same policy for all machines, as opposed to [1]. SPT and LPT are policies that schedule the jobs without preemption respectively in order of increasing or decreasing processing times with a deterministic tie-breaking rule for each machine. An interesting property of SPT is that it minimizes the sum of the completion times, while LPT has a better price of anarchy, because it incites small jobs to go on the least loaded machine which smoothes the loads. A policy that relates individual costs even stronger to the social cost is MAKESPAN, where jobs are scheduled in parallel on one machine using time-multiplexing and assigned each job a fraction of the CPU that is proportional to its processing time. As a result all jobs complete at the same time, and the individual cost is the load of the machine.

What could a scheduler do in the non-clairvoyant case? He could either schedule the jobs in a random order or in parallel. The RANDOM policy schedules the jobs in a random order without preemption. Consider a job i assigned to machine j in the schedule σ, then the cost of i under the RANDOM policy is its expected completion time [21], i.e.,

$$c_i = p_{ij} + \frac{1}{2} \sum_{i':\sigma(i')=j,\ i'\neq i} p_{i'j}.$$

In other words the expected completion time of i is half of the total load of the machine, where job i counts twice. Again, as for MAKESPAN, the individual and social cost in RANDOM are strongly related, and it is likely that these policies should have the same price of anarchy. That is is indeed the case except for unrelated machines.

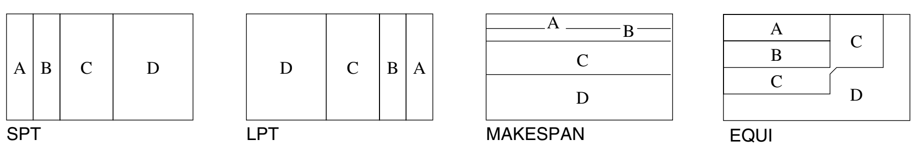


Fig. 1. Illustration of different scheduling policies for $p_A = 1, p_B = 1, p_C = 2, p_D = 3$. Tie is broken arbitrarily between jobs A and B. The rectangles represent the schedules on a single machine with time going from left to right and the hight of a block being the amount of CPU assigned to the job.

Another natural non-clairvoyant policy is EQUI. As MAKESPAN it schedules the jobs in parallel preemptively using time-multiplexing and assigns to every job the same fraction of the CPU. Suppose there are k jobs with processing times $p_{1j} \leq p_{2j} \leq \ldots \leq p_{kj}$ assigned to machine j, we renumbered jobs from 1 to k for this example. Since, each job receives the same amount of resource, then job 1 is completed at time $c_1 = kp_{1j}$. At that time, all jobs have remaining processing time $(p_{2j} - p_{1j}) \leq (p_{3j} - p_{1j}) \leq \ldots \leq (p_{kj} - p_{1j})$. Now the machine splits its resource into $k-1$ parts until the moment job 2 is completed, which is at $kp_{1j} + (k-1)(p_{2j} - p_{1j}) = p_{1j} + (k-1)p_{2j}$. In general, the completion time of job i, which is also its cost, under EQUI policy is:

$$c_i = c_{i-1} + (k-i+1)(p_{ij} - p_{i-1,j}) \quad (1)$$
$$= p_{1j} + \ldots + p_{i-1,j} + (k-i+1)p_{ij} \quad (2)$$

We already distinguished policies depending on what information is needed from the jobs. In addition we distinguish between *preemptive* and *non-preemptive* policies, depending on the schedule that is produced. Among the policies we considered so far, only MAKESPAN and EQUI are preemptive, in the sense that they rely on time-multiplexing, which consists in executing arbitrary small slices of the jobs. Note that, EQUI is a realistic and quite popular policy. It is implemented in many operating systems such as Unix, Windows.

1.2 Previous and Related Work

Coordination mechanism are related to local search algorithms. The local improvement moves in the local search algorithm correspond to the best-response moves of players in the game defined by the coordination mechanism. Some results on local search algorithms for scheduling problem are surveyed in [24].

Most previous work concerned non-preemptive strongly local policies, in particular the MAKESPAN policy. [10] gave tight results $\Theta(\log m / \log\log m)$ of its price of anarchy for pure Nash equilibria on uniform machines. [14] extended this result for the strong price of anarchy, and obtained the tight bound $\Theta(\log m/(\log\log m)^2)$. In addition, [17] and [3] gave tight bounds for the price of anarchy for restricted identical machines.

Coordination mechanism design was introduced by [9]. They studied the LPT policy on identical machines. [21] studied coordination mechanism for all four machine environments and gave a survey on the results for non-preemptive strongly local policies. They also analyzed the existence of pure Nash equilibria under SPT, LPT and RANDOM for certain machine environments and the speed of convergence to equilibrium of the best response dynamics. Precisely, they proved that the game is a potential game under the policies SPT on unrelated machines, LPT on uniform or restricted identical machines, and RANDOM on restricted identical machines. The policy EQUI has been studied in [13] for its competitive ratio. The results are summarized in Table 1.

For local policies, [5] introduced the inefficiency-based policy which has price of anarchy $O(\log m)$ on unrelated machines and modified it to get a policy which always admits an equilibrium and the price of anarchy is $O(\log^2 m)$. Moreover, they also proved that every non-preemptive strongly local policy with an additional assumption has price of anarchy at least $m/2$, which shows a sharp difference between strongly local and local policies. [7] gave three local policies with price of anarchy $O(\log m)$, $O(\log m/\log\log m)$ and $O(\log^2 m)$, respectively, in which the games under first and the third policies always admit an equilibrium.

Table 1. Price of anarchy under different strongly local and non-clairvoyant policies. The right most column is a contribution of this paper, which is also the strong price of anarchy.

model \ policy	MAKESPAN	SPT	LPT	RANDOM	EQUI
identical	$2-\frac{2}{m+1}$ [15, 23]	$2-\frac{1}{m}$ [18, 21]	$\frac{4}{3}-\frac{1}{3m}$ [19, 9]	$2-\frac{2}{m+1}$ [15, 23]	$2-\frac{1}{m}$
uniform	$\Theta(\frac{\log m}{\log\log m})$ [10]	$\Theta(\log m)$ [2, 21]	$1.52 \le PoA \le 1.59$ [11, 16, 21]	$\Theta(\frac{\log m}{\log\log m})$ [10]	$\Theta(\log m)$
restricted id.	$\Theta(\frac{\log m}{\log\log m})$ [17, 3]	$\Theta(\log m)$ [2, 21]	$\Theta(\log m)$ [4, 21]	$\Theta(\frac{\log m}{\log\log m})$ [17, 3]	$\Theta(\log m)$
unrelated	unbounded [23]	$\Theta(m)$ [8, 20, 5]	unbounded	$\Theta(m)$ [21]	$\Theta(m)$

1.3 Our Contribution

We are interested in *admissible* non-clairvoyant policies – policies that always induce a Nash equilibrium for any instance of the game. Maybe more important than the question of existence of Nash equilibrium is the question of convergence to an equilibrium. Since no processing times are known to the coordination mechanism, it is impossible to compute some equilibria. As all processing times are known to all jobs, it makes sense to let the jobs evolve according to the best-response dynamics, until they eventually reach an equilibria. Therefore it is important to find out under which conditions the dynamics converges.

For the unrelated machine model, we call a job i *balanced* if the ratio of its processing times is bounded by 2, meaning $\max_j p_{ij} / \min_j p_{ij} \le 2$. In addition for the uniform machine model, we say that machines have *balanced speeds* if the maximum and minimum speeds differ at most by factor 2. Note that, in the model of uniform machines with balanced speeds, jobs are all balanced.

In Section 2, we study the existence of Nash equilibrium under the non-clairvoyant policies RANDOM and EQUI. We show that in the RANDOM policy, the game always possesses an equilibrium on uniform machines with speed ratio at most 2. We also show that on two unrelated machines, it is a potential game, but for three unrelated machines or more the best-response dynamic does not converge. These results partly answer open questions in [21]. Moreover, we prove that for the EQUI policy, the game is a (strong) potential game, see Table 2. Note that, in our proofs of equilibrium existence, it is sufficient to show that there exist potential functions which are strictly decreased at each step of the dynamic. As the game is finite, after finite number of steps, the potential function will converge to an equilibrium.

In Section 3, we analyze the price of anarchy and the strong price of anarchy of EQUI in different machine models. In uniform and restricted identical machine models, our main contributions are the lower bounds of the price of anarchy. Moreover, we prove that these lower bounds also hold for the strong price of anarchy. We observe that, except for these models, RANDOM is slightly better than EQUI. In the unrelated machine model, interestingly, the price of anarchy of

Table 2. Convergence of the best response dynamic and existence of equlibria. (*) A refinement of the best response dynamic converges when machines have balanced speeds. (**) The best-response dynamic does not converge for $m \geq 3$ machines, but converges for $m = 2$ machines and balanced jobs.

<table>
<tr><th>model \ policy</th><th>MAKESPAN</th><th>SPT</th><th>LPT</th><th>RANDOM</th><th>EQUI</th></tr>
<tr><td>identical</td><td rowspan="4">Yes [folklore]</td><td rowspan="4">Yes [21]</td><td rowspan="3">Yes [21]</td><td>Yes [21]</td><td rowspan="4">Yes</td></tr>
<tr><td>uniform</td><td>(*)</td></tr>
<tr><td>restricted identical</td><td>Yes [21]</td></tr>
<tr><td>unrelated</td><td>No</td><td>(**)</td></tr>
</table>

EQUI reaches the lower bound in [5] on the PoA of any *non-preemptive* strongly local policy with some additional condition. The latter showed that although there is a clear difference between strongly local and local policies with respect to the price of anarchy, our result indicates that in contrast, restricting strongly local policies to be non-clairvoyant does not really affect the price of anarchy. Moreover, EQUI policy does not need any knowledge about jobs' characteristics, even their identities (IDs) which are useful in designing policies with low price of anarchy in [5, 7].

Due to the limit of space, only some proofs are presented in the paper. The others can be found in the full version [12].

2 Existence of Nash Equilibrium

Summary of Results on the Existence of Nash Equilibrium. *We consider the scheduling game under different policies in different machine environments.*

1. *For the RANDOM policy on unrelated machines, it is not a potential game for 3 or more machines, but it is a potential game for 2 machines and balanced jobs. On uniform machines with balanced speeds, the RANDOM policy induces a Nash equilibrium.*
2. *For the EQUI policy it is an exact potential game.*

2.1 The RANDOM Policy for Unrelated Machines

In the RANDOM policy, the cost of a job is its expected completion time. If the load of machine j is ℓ_j then the cost of job i assigned to machine j is $\frac{1}{2}(\ell_j + p_{ij})$. We see that a job i on machine j has an incentive to move to machine j' if and only if $p_{ij} + \ell_j > 2p_{ij'} + \ell_{j'}$. In the following, we will characterize the game under the RANDOM policy in the unrelated model as a function of the number of machines.

Theorem 1. *The game is a potential game under the RANDOM policy on 2 machines with balanced jobs.*

Lemma 1. *The best-response dynamic does not converge under the RANDOM policy on 3 or more machines.*

2.2 The RANDOM Policy on Uniform Machines with Balanced Speeds

Let $p_1 \le p_2 \le \ldots \le p_n$ be the job lengths and $s_1 \ge s_2 \ge \ldots \ge s_m$ be the machine speeds. Now the processing time of job i on machine j is p_i/s_j. A *new unhappy* job with respect to a move is a job that was happy before the move and has become unhappy by this move.

Lemma 2. *Consider a job i making a best move from machine a to b on uniform machines with balanced speeds. We have that if there is a new unhappy job with index greater than i then $s_a > s_b$.*

Theorem 2. *On uniform machines with balanced speeds, there always exist Nash equilibria under the RANDOM policy.*

Proof. We use a potential argument on a refinement of the best-response dynamic. Consider a best-response dynamic in which among all unhappy jobs, the one with the greatest index makes the best move. By numbering convention, this job has the greatest length among all unhappy jobs. Given a strategy profile σ, let t be the unhappy job of greatest index. We encode t by a characteristic function $f_\sigma : \{1, 2, \ldots, n\} \to \{0, 1\}$ as $f_\sigma(i) = 1$ if $1 \le i \le t$, otherwise $f_\sigma(i) = 0$. Define the potential function $\Phi(\sigma) = (f_\sigma(1), s_{\sigma(1)}, f_\sigma(2), s_{\sigma(2)}, \ldots, f_\sigma(n), s_{\sigma(n)})$. We claim that in each step of the best-response dynamic described above, the potential function decreases strictly lexicographically.

Let t be the unhappy job of greatest index in the strategy profile σ, let t' be the unhappy job of greatest index in σ' – the strategy profile after the move of t. Note that the unique difference between σ and σ' is the machine to which job t moved. If $t' < t$, we have that $f_\sigma(i) = f_{\sigma'}(i) = 1$, $s_{\sigma(i)} = s_{\sigma'(i)}$ for all $i \le t'$ and $1 = f_\sigma(t'+1) > f_{\sigma'}(t'+1) = 0$. Thus, $\Phi(\sigma) > \Phi(\sigma')$. If $t' > t$, we also have that $f_\sigma(i) = f_{\sigma'}(i) = 1$, $s_{\sigma(i)} = s_{\sigma'(i)}$ for all $i < t$ and $f_\sigma(t) = f_{\sigma'}(t) = 1$. However, $t' > t$ means that there are some new unhappy jobs with lengths greater than p_t, hence by Lemma 2, $s_{\sigma(t)} > s_{\sigma'(t)}$. In the other words, $\Phi(\sigma) > \Phi(\sigma')$. Therefore, the dynamic converges, showing that there always exists a Nash equilibrium. □

2.3 The EQUI Policy

In the EQUI policy, the cost of job i assigned to machine j is given by formulation in (1) and (2). Here is an alternative formulation for the cost

$$c_i = \sum_{\substack{i':\sigma(i')=j \\ p_{i'j} \le p_{ij}}} p_{i'j} + \sum_{\substack{i':\sigma(i')=j \\ p_{i'j} > p_{ij}}} p_{ij}$$

Theorem 3. *The game with the EQUI policy is an exact potential game. In addition, it is a strong potential game, in the sense that the best-response dynamic converges even with deviations of coalitions.*

3 Inefficiency of Equilibria under the EQUI Policy

In this section, we study the inefficiency of the game under the EQUI policy which is captured by the price of anarchy (PoA) and the strong price of anarchy (SPoA). Note that the set of strong Nash equilibria is a subset of that of Nash equilibria so the SPoA is at most as large as the PoA. We state the main theorem of this section. Whenever we bound *(S)PoA* we mean that the bound applies to both the price of anarchy and the strong price of anarchy.

Summary of Results on the Price of Anarchy. *The game under the EQUI policy has the following inefficiency.*

1. *For identical machines, the (S)PoA is* $2 - \frac{1}{m}$.
2. *For uniform machines, the (S)PoA is* $\Theta(\min\{\log m, r\})$ *where* r *is the number of different machine's speeds in the model.*
3. *For restricted identical machines, the (S)PoA is* $\Theta(\log m)$.
4. *For unrelated machines, the (S)PoA is* $\Theta(m)$.

In the following, we concentrate on the inefficiency of equilibria in unrelated machines. We prove that the PoA of the game under the EQUI policy is upper bounded by $2m$. Interestingly, without any knowledge of jobs' characteristics, the inefficiency of EQUI – a non-clairvoyant policy – is the same up to a constant compared to that of SPT – the best strongly local policy with price of anarchy $\Theta(m)$.

Theorem 4. *For unrelated machines, the price of anarchy of policy EQUI is at most* $2m$.

Proof. For job i, let q_i be the smallest processing time of i among all machines, i.e., $q_i := \min_j p_{ij}$ and let $Q(i) := \text{argmin}_j p_{ij}$ be the corresponding machine. Without loss of generality we assume that jobs are indexed such that $q_1 \leq q_2 \leq \ldots \leq q_n$. Note that $\sum_{i=1}^{n} q_i \leq m \cdot OPT$, where OPT is the optimal makespan, as usual. First, we claim the following lemma.

Lemma 3. *In any Nash equilibrium, the cost* c_i *of job* i *is at most*

$$2q_1 + \ldots + 2q_{i-1} + (n - i + 1)q_i. \tag{3}$$

Proof. The proof is by induction on i. The cost of job 1 on machine $Q(1)$ would be at most nq_1, simply because there are at most n jobs on this machine. Therefore the cost of job 1 in the Nash equilibrium is also at most nq_1. Assume the induction hypothesis holds until index $i - 1$. Consider job i. Since the strategy profile is a Nash equilibrium, i's current cost is at most its cost if moving to machine $Q(i)$. We distinguish different cases. In these cases, denote c'_i as the new cost of i if it moves to machine $Q(i)$.

1. **All Jobs t Scheduled on Machine $Q(i)$ Satisfy $t > i$.**
   This case is very similar to the basis case. There are at most $n - i$ jobs on machine $Q(i)$, beside i. The completion time of job i is then at most $(n-i+1)q_i$ which is upper bounded by (3). For the remaining cases, we assume that there is a job $i' < i$ scheduled on $Q(i)$.

2. **There is a Job $t < i$ on Machine $Q(i)$ Such that $p_{tQ(i)} \geq p_{iQ(i)} (= q_i)$.** Since $p_{tQ(i)} \geq q_i$, the new cost of job i is not more than the new cost of job t. Moreover, the new cost of job t is increased by exactly q_i, so the new cost of i is bounded by

$$\begin{aligned} c'_i &\leq c_t + q_i \\ &\leq 2q_1 + \ldots + 2q_{t-1} + (n-t+1)q_t + q_i \\ &= 2q_1 + \ldots + 2q_{t-1} + 2(i-t)q_t + (n-2i+t+1)q_t + q_i \\ &\leq 2q_1 + \ldots + 2q_{t-1} + 2q_t + \ldots + 2q_{i-1} + (n-i+1)q_i, \end{aligned}$$

where the first inequality uses the induction hypothesis and the last inequality is due to $t < i$ and $q_t \leq q_{t+1} \leq \ldots \leq q_i$.
3. **Every Job t Scheduled on Machine $Q(i)$ with $p_{tQ(i)} \geq q_i$ Satisfies $t \geq i$.** Since we are not in the first two cases, there is a job $t < i$ on machine $Q(i)$ with $p_{tQ(i)} < q_i$. Let i' be the job of greatest index among all jobs scheduled on $Q(i)$ with smaller processing time than q_i. All jobs t scheduled on $Q(i)$ and having smaller processing time than that of i, also have smaller index because $q_t \leq p_{tQ(i)} \leq q_i$. Therefore i' is precisely the last job to complete before i. At the completion time of i' there are still $q_i - p_{i'Q(i)} \leq q_i - q_{i'}$ units of i to be processed. By the case assumption, there are at most $(n-i)$ jobs with processing time greater than that of i. Therefore the new cost of i is at most

$$\begin{aligned} c'_i &= c_{i'} + (n-i+1)(q_i - q_{i'}) \\ &\leq 2q_1 + \ldots + 2q_{i'-1} + (n-i'+1)q_{i'} + (n-i+1)(q_i - q_{i'}) \\ &= 2q_1 + \ldots + 2q_{i'-1} + (i-i')q_{i'} + (n-i+1)q_i \\ &\leq 2q_1 + \ldots + 2q_{i'-1} + (q_{i'} + \ldots + q_{i-1}) + (n-i+1)q_i \\ &\leq 2q_1 + \ldots + 2q_{i-1} + (n-i+1)q_i \end{aligned}$$

where the first inequality uses the induction hypothesis and the third inequality is due to the monotonicity of the sequence $(q_j)_{j=1}^n$. □

Since the term $2q_1 + \ldots + 2q_{i-1} + (n-i+1)q_i$ is increasing in i and at $i = n$ this term is $2\sum_{i=1}^n q_i \leq 2m \cdot OPT$, the cost of each job in an equilibrium is bounded by $2m \cdot OPT$, so the price of anarchy is at most $2m$. □

We provide a game instance, inspired by the work of [5], showing that the upper bound analyzed above is tight.

Lemma 4. *The (strong) price of anarchy of* EQUI *is at least* $(m+1)/4$.

Acknowledgments. We would like to thank Adi Rosén for helpful discussions.

References

[1] Angel, E., Bampis, E., Pascual, F.: Truthful algorithms for scheduling selfish tasks on parallel machines. Theoretical Computer Science (TCS) 369, 157–168 (2006)

[2] Aspnes, J., Azar, Y., Fiat, A., Plotkin, S.A., Waarts, O.: On-line routing of virtual circuits with applications to load balancing and machine scheduling. Journal of the ACM 44(3), 486–504 (1997)

[3] Awerbuch, B., Azar, Y., Richter, Y., Tsur, D.: Tradeoffs in worst-case equilibria. Theoretical Computer Science 361(2-3), 200–209 (2006)

[4] Azar, Y., Naor, J., Rom, R.: The competitiveness of on-line assignments. Journal of Algorithms 18(2), 221–237 (1995)

[5] Azar, Y., Jain, K., Mirrokni, V.S.: (Almost) Optimal Coordination Mechanisms for Unrelated Machine Scheduling. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 323–332 (2008)

[6] Brucker, P.: Scheduling Algorithms, 3rd edn. Springer, Heidelberg (2001)

[7] Caragiannis, I.: Efficient coordination mechanisms for unrelated machine scheduling. In: SODA, pp. 815–824 (2009)

[8] Cho, Y., Sahni, S.: Bounds for list schedules on uniform processors. SIAM Journal on Computing 9, 91–103 (1980)

[9] Christodoulou, G., Koutsoupias, E., Nanavati, A.: Coordination mechanisms. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 345–357. Springer, Heidelberg (2004)

[10] Czumaj, A., Vöcking, B.: Tight bounds for worst-case equilibria. In: Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 413–420 (2002)

[11] Dobson, G.: Scheduling independent tasks on uniform processors. SIAM Journal on Computing 13, 721–726 (1984)

[12] Durr, C., Kim, T.N.: Non-clairvoyant scheduling games, http://www.lix.polytechnique.fr/~thang/Papers/EQUI.pdf

[13] Edmonds, J.: Scheduling in the dark. In: Proceedings of the 31st ACM Symposium on Theory of Computing, STOC (1999)

[14] Fiat, A., Kaplan, H., Levy, M., Olonetsky, S.: Strong Price of Anarchy for Machine Load Balancing. In: Proceedings of the 34th International Colloquium on Automata, Languages and Programming, pp. 583–594 (2007)

[15] Finn, G., Horowitz, E.: A linear time approximation algorithm for multiprocessor scheduling. BIT 19, 312–320 (1979)

[16] Friesen, D.K.: Tighter bounds for LPT scheduling on uniform processors. SIAM Journal on Computing 16, 554–560 (1987)

[17] Gairing, M., Lücking, T., Mavronicolas, M., Monien, B.: Computing Nash equilibria for scheduling on restricted parallel links. In: 36th ACM Symposium on Theory of Computing, pp. 613–622 (2004)

[18] Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell System Technical Journal 45, 1563–1581 (1966)

[19] Graham, R.L.: Bounds on multiprocessing timing anomalies. SIAM Journal of Applied Mathematics 45, 416–429 (1969)

[20] Ibarra, O.H., Kim, C.E.: Heuristic algorithms for scheduling independent tasks on nonidentical processors. Journal of the ACM 24, 280–289 (1977)

[21] Immorlica, N., Li, L., Mirrokni, V.S., Schulz, A.: Coordination Mechanisms for Selfish Scheduling. In: Proceedings of the 1st International Workshop on Internet and Network Economics, pp. 55–69 (2005)
[22] Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14, 124–143 (1996)
[23] Schuurman, P., Vredeveld, T.: Performance guarantees of local search for multiprocessor scheduling. Informs Journal on Computing 361(1), 52–63 (2007)
[24] Vredeveld, T.: Combinatorial Approximation Algorithms: Guaranteed Versus Experimental Performance. PhD thesis, Technische Universiteit Eindhoven, The Netherlands (2002)

The Balloon Popping Problem Revisited: Lower and Upper Bounds

Hyunwoo Jung and Kyung-Yong Chwa

Division of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{solarity,kychwa}@jupiter.kaist.ac.kr

Abstract. We consider the balloon popping problem introduced by Immorlica et al. in 2007 [13]. This problem is directly related to the problem of profit maximization in online auctions, where an auctioneer is selling a collection of identical items to anonymous unit-demand bidders. The auctioneer has the full knowledge of bidders' private valuations for the items and tries to maximize his profit. Compared with the profit of fixed price schemes, the competitive ratio of Immorlica et al.'s algorithm was in the range $[1.64, 4.33]$. In this paper, we narrow the gap to $[1.659, 2]$.

Keywords: auction, lower bound, upper bound.

1 Introduction

In auctions, sellers try to maximize profits by selling items at high prices. Buyers have individual valuations for items and try to buy items at low prices relative to their valuations. We can categorize auction problems into two branches: profit maximization problems and social welfare maximization problems. Profit maximization problems focus on maximizing seller's profit [1,3,6,12]. Social welfare maximization problems concentrate on maximizing the sum of buyers' valuations [7,8,10,16].

In online auctions, the auctioneer allocates items to bidders when bidders or items arrive online. In recent years, many online and offline auction algorithms were designed, and their performance was analyzed using competitive analysis [2,6,5,13,14,15]. To analyze competitive ratio, usually the profit of an online auction is compared with that of fixed price mechanisms [3,6,5,4].

In this paper, we consider the problem of profit maximization in the following auction setting. The seller possesses a number of identical items, and the number of buyers is equal to the number of items. The seller repeatedly announces a price, and each buyer decides whether to stay or leave. The seller can sell an item at any time to each buyer at a price accepted by the buyer. Each buyer can receive at most one item. We assume a special distribution for the buyers' valuations, and compare the power of an auction with that of a fixed price mechanism.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 147–158, 2009.

1.1 Previous Results and Our Results

Immorlica et al. [13] studied the power of ascending auctions in a scenario in which the seller sells multiple identical items to anonymous unit-demand bidders. They showed that even with full knowledge of bidders' private valuations, if the bidders are ex-ante identical, no ascending auction can get profit more than a constant times the profit of the best fixed price scheme. Our work in this paper narrows the gap proved by Immorlica et al. [13]. Although much of the work in auctions concentrates on design of truthful mechanisms [18], in this paper, we assume that bidders are truthful. Also, we assume a discrete distribution for bidders' valuations.

We describe the balloon popping problem and online balloon popping mechanisms in Section 2. In Section 3, we give an improved lower bound for the balloon popping problem. In Section 4, we derive a variant of ballot theorem which can be applied to our problem. In Section 5, we give an improved upper bound for the online balloon popping mechanism. Finally, we conclude with remarks.

2 Problem Description

Immorlica et al. [13] formulate their ascending auction problem as a balloon popping problem as follows.

The Balloon Popping Problem. Suppose that n indistinguishable balloons, yet to be blown up, are given. Each balloon has an allowable capacity. If a balloon is blown up more than the capacity, then it is popped. Suppose also that the set of capacities of the given n balloons is $\{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$. The objective is to maximize the sum of the volumes of the blown-up balloons.

To get a feeling for the problem, consider the following strategy.

strategy **inc**:

- Initialize i to be $n-1$. Blow up all the balloons to $\frac{1}{n}$.
- While $i \neq 0$, do the following: Select a balloon among the unpopped and the unselected balloons uniformly at random. Blow up all the unpopped and unselected balloons to $\frac{1}{i}$. If there is a balloon which is popped while being blown up, deselect the balloon selected just before.
- $i := i - 1$.

The strategy **inc** is ascending auction-style, unlike the strategy in Immorlica et al., which is descending auction-style. Under strategy **inc**, the balloon with capacity $\frac{1}{j}$ will be unpopped if it is selected in the iteration of the strategy **inc** when there are j unpopped balloons remaining. Indeed, before this iteration, the balloon with capacity $\frac{1}{j}$ is not popped. Thus, the probability that this balloon is not popped is $\frac{1}{j}$, and its volume, when blown up, is $\frac{1}{j}$. Hence, the expected final total volume is $\sum_{1 \leq k \leq n} \frac{1}{k^2}$, which goes to $\frac{\pi^2}{6} \approx 1.64$ when $n \to \infty$.

In the balloon popping problem, the balloons correspond to bidders in an online auction problem, and the capacities of the balloons are bidders' private valuations. The objective corresponds to maximizing profits of the auctioneer.

Let $\mathbf{B_n}$ be the optimal expected volume for the balloon popping problem with n balloons. Let $\mathbf{B}_\infty = \lim_{n\to\infty} \mathbf{B_n}$. We will refer to the balloon with capacity $\frac{1}{i}$ as the $\frac{1}{i}$-balloon.

Immorilica et al. defined an online balloon popping mechanism corresponding to an online auction. The online mechanism processes balloons sequentially in a fixed order. It is assumed that if a balloon has been processed, its capacity is revealed irrespective of whether this balloon was popped. We reproduce the definition of Immorlica et al. below.

Online Balloon Popping Mechanism. Let us assume that we are given balloons with capacities $v_1 \geq v_2 \geq \cdots \geq v_n$. Balloons are ordered by a random permutation π, i.e., the i-th balloon has capacity v_{π_i}. An online balloon popping mechanism is defined by a function $Blow(v_{\pi_1}, \cdots, v_{\pi_{i-1}})$ that outputs a non-negative number b, indicating that b units of air should be blown into balloon i. If $b \leq v_{\pi_i}$, let $s_i = b$. Otherwise, let $s_i = 0$. The payoff or profit of the online balloon popping mechanism is $\sum_i s_i$.

Immorlica et al. [13] proved that we can assume without loss of generality the set of bidders' valuations (balloon capacities) is $\{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$.

Lemma 1. [13] *Suppose that the capacities of the balloons are* $v_1 \geq v_2 \geq \cdots \geq v_n$*. Without loss of generality, assume that* $\max_i iv_i = 1$*. Then the maximum expected volume achievable by a balloon popping mechanism on* $\{v_1, v_2, \cdots, v_n\}$ *is at most the maximum achievable on* $\{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$*.*

Denote the optimal value of an online balloon popping mechanism for n balloons by $\mathbf{ONOPT_n}$. Immorlica et al. proved that $\mathbf{B_n} \leq \mathbf{ONOPT_n}$ by reducing the balloon popping problem to designing an online balloon popping mechanism.

Theorem 1. [13] $\mathbf{B_n} \leq \mathbf{ONOPT_n}$.

Also, Immorlica et al. characterize the profit of the optimal online balloon popping mechanism by the following simple formula.

Theorem 2. [13] *Let T be a random subset of $\{v_1, \ldots, v_n\}$, and let $\frac{1}{y_1}, \frac{1}{y_2}, \cdots, \frac{1}{y_{|T|}}$ be the order statistics of the set T (so $\frac{1}{y_1}$ is the largest element of T, $\frac{1}{y_2}$ is the second largest element of T, etc.). Define the random variable $g(T)$ to be $g(T) = \frac{1}{|T|} max_{j=1,\ldots,|T|} \frac{j}{y_j}$. Then the revenue of the optimal online balloon popping mechanism is given by $\sum_{k=1}^{n} E_T[g(T)]$, where the expectation in the k-th term in the summation is over a random subset T of $\{v_1, \ldots, v_n\}$ of size k.*

Immorlica et al. proved that $\frac{\pi^2}{6} \leq \mathbf{B}_\infty \leq 4.3331$. We improve both the lower and the upper bound on $\mathbf{B}_\infty$. The following is the main theorem in this paper.

Theorem 3. *For the balloon popping problem with n balloons,* $1.659 \leq \mathbf{B}_\infty \leq 2$.

Together, Theorem 3 and Lemma 1 imply the following theorem.

Theorem 4. *For balloon capacities $v_1 \geq v_2 \geq \cdots \geq v_n$, no balloon popping strategy can achieve an expected volume that exceeds* $2 \cdot \max_i iv_i$.

In order to prove the upper bound of $\mathbf{B_n}$, in Section 4 we consider an extension of the famous generalized ballot problem.

3 Lower Bound

In this section, we suggest a simple algorithm for the balloon popping problem that achieves a profit of at least 1.659.

We assume that the balloons are given in the order of $b_1, b_2, \ldots, b_n$, where $\{b_1, b_2, \ldots, b_n\} = \{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$.

In the following algorithm **Bunch**, V means the set of balloon capacities still under consideration. Intuitively speaking, in the first **while** loop of **Bunch**, we process balloons one by one, trying to fill them to the maximum possible capacity. We stop when all balloons with capacities between $\frac{1}{y}$ and 1 are either filled or popped. In the second **while** loop of **Bunch**, we deal with sets of z consecutive balloons, finishing each set before beginning the next.

Algorithm 1. Bunch

$V \Leftarrow \{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$
$i \Leftarrow 1$
$v \leftarrow max(V)$
Fix integer constants y and z such that $y < n$ and $z < n$.
while v is greater than or equal to $\frac{1}{y}$ **do**
  Blow the balloon b_i up to the volume v
  if the balloon b_i pops at a volume $\frac{1}{k}$ **then**
    $V \Leftarrow V - \{\frac{1}{k}\}$
    $s_i \Leftarrow 0$
  else
    $V \Leftarrow V - \{v\}$
    $s_i \Leftarrow v$
  end if
  Increase i by one
  $v \leftarrow max(V)$
end while
while $i <= n$ **do**
  $j \Leftarrow$ the minimum integer such that $zj \geq \frac{1}{max(V)}$.
  $c \Leftarrow$ the maximum integer such that $c \leq zj$ and $\frac{1}{c} \in V$.
  while there exists a balloon with volume at least $\frac{1}{c}$ among the remaining balloons $b_i, b_{i+1}, \ldots, b_n$ **do**
    Blow the balloon b_i up to the volume $\frac{1}{c}$
    if the balloon pops at a volume $\frac{1}{k}$ **then**
      $V \Leftarrow V - \frac{1}{k}$
      $s_i \Leftarrow 0$
    else
      $s_i \Leftarrow \frac{1}{c}$
    end if
    Increase i by one
  end while
  Remove from V all values that are greater than or equal to $\frac{1}{c}$
end while

After the algorithm **Bunch** terminates, the total volume is $\sum_{1\le i\le n} s_i$. For example, let us fix $n = 7$, $y = 2$, and $z = 2$. Consider a collection of balloons with capacities $\frac{1}{2}, \frac{1}{4}, 1, \frac{1}{3}, \frac{1}{6}, \frac{1}{5}, \frac{1}{7}$ (in this order). The algorithm **Bunch** blows the 1st balloon up to 1. The 1st balloon pops, so $s_1 = 0$. Then the 2nd balloon is blown up to 1. It also pops, so $s_2 = 0$. The 3rd balloon is blown up to 1, and does not pop, so $s_3 = 1$. Next, the 4th balloon is blown up to $\frac{1}{3}$, so $s_4 = \frac{1}{3}$. The 5th balloon is blown up to $\frac{1}{6}$, so $s_5 = \frac{1}{6}$. The 6th balloon is blown up to $\frac{1}{6}$, so $s_6 = \frac{1}{6}$. Finally, the 7th balloon is blown up to $\frac{1}{7}$, so $s_7 = 0$. The final volume in this case is $1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{6} + \frac{1}{7}$. In general, the algorithm **Bunch** with $y = 2$ and $z = 2$ gives us expected profit of 1.6596.

Lemma 2. *When $y = 2$ and $z = 2$, algorithm **Bunch** has expected profit that approaches* 1.6596 *when $n \to +\infty$.*

Proof. The expected total profit is $E[\sum_{1\le i\le n} V_i]$, where V_i denotes the volume of the $\frac{1}{i}$-balloon. We have $E[V_1] = 1$, $E[V_2] = \frac{1}{2^2}$. For $i \ge 2$, the $\frac{1}{2i-1}$-balloon can have non-zero volume $\frac{1}{2i-1}$ or $\frac{1}{2i}$. To get non-zero volume, it must not pop, i.e., it has to appear later than the balloons with capacities $1, \frac{1}{2}, \ldots, \frac{1}{2i-2}$. Further, it gets volume $\frac{1}{2i-1}$ if the $\frac{1}{2i}$-balloon pops before the $\frac{1}{2i-1}$-balloon appears, which means that the $\frac{1}{2i}$-balloon must appear before one of the balloons $1, \frac{1}{2}, \ldots, \frac{1}{2i-2}$. Thus, the probability that the $\frac{1}{2i-1}$-balloon gets volume $\frac{1}{2i-1}$ is $\frac{1}{2i-1}\frac{2i-2}{2i}$. Similarly, the probability that it gets volume $\frac{1}{2i}$ is $\frac{1}{2i-1}\frac{2}{2i}$. Hence, for $i \ge 2$, we have $E[V_{2i-1}] = \frac{1}{2i-1}\frac{1}{2i-1}\frac{2i-2}{2i} + \frac{1}{2i}\frac{1}{2i-1}\frac{2}{2i}$.

By the algorithm **Bunch**, the $\frac{1}{2i}$-balloon can get non-zero volume $\frac{1}{2i}$. This can only happen if this balloon appears after the balloons with capacities $1, \frac{1}{2}, \ldots, \frac{1}{2i-2}$, which happens with probability $\frac{1}{2i-1}$. Thus, for $i \ge 2$, we have $E[V_{2i}] = \frac{1}{2i}\frac{1}{2i-1}$. Further, it can be shown that $\lim_{n\to+\infty}\sum_{2\le i\le\infty} E[V_{2i-1}] = -\frac{1}{2} + 4\ln(2) - \frac{5}{24}\pi^2$, and $\lim_{n\to+\infty}\sum_{2\le i\le\infty} E[V_{2i}] = -\frac{1}{2} + \ln(2)$. Therefore, $E[\sum_{1\le i\le\infty} V_i] \approx 1.6596$. □

We obtain the following corollary.

Corollary 1. $1.659 \le \mathbf{B}_\infty$.

4 A Ballot Theorem

In this section, we derive a version of the ballot theorem to prove an upper bound of B_∞. The generalized ballot problem is stated as follows.

The Ballot Problem. Suppose that in an election, candidate A receives n votes and candidate B receives k votes, where $n \ge sk$ for some positive integer s. Compute the number of ways that the ballots can be ordered so that A maintains more than s times as many votes as B does throughout the counting of the ballots.

The solution to this problem is well-known.

The ballot theorem. The solution to the ballot problem is $\frac{n-sk}{n+k}\binom{n+k}{k}$.

The original ballot problem corresponds to the case $s = 1$. For integer $s \geq 1$, Barbier has generalized the problem without proof. Later various proofs for the ballot problem has been found [17]. In this paper we use a cycle lemma [9] to prove a variant of the ballot theorem.

The ballot problem can be interpreted as counting the lattice paths from $(0,0)$ to (k,n) that do not touch the line $y = sx$ after the point $(0,0)$ and before (k,n). A lattice path from $(0,0)$ to (k,n) can be seen as a sequence $p^0p^1p^2\cdots p^t$ of $n+k+1$ vertices, where $p^0 = (0,0)$ and $p^t = (k,n)$. From now on, "lattice path" always refers to a lattice path from $(0,0)$ to (k,n). Define an *x-strict lattice path* as a lattice path such that $p^1 = (0,1)$ and no three consecutive vertices have the same y-coordinates (see Figure 1). Note that an x-strict lattice path can be represented by a sequence of k horizontal segments with different y-coordinates and n vertical segments. Since there are n possibilities for the y-coordinates of the k horizontal segments, the number of x-strict lattice paths from $(0,0)$ to (k,n) is $\binom{n}{k}$. Let $\mathcal{A}$ be the set of all x-strict lattice paths from $(0,0)$ to (k,n). In this section we prove the following theorem.

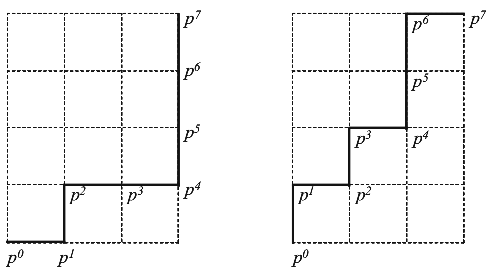


Fig. 1. The left figure is a non x-strict lattice path from (0,0) to (3,4). The right figure is an x-strict lattice path from (0,0) to (3,4).

Theorem 5. *The number of x-strict lattice paths not touching the line $y = \frac{n}{k}x$ is at most $\frac{1}{k}\binom{n-1}{k-1}$. Moreover, when n and k are relatively prime, it equals to $\frac{1}{k}\binom{n-1}{k-1}$.*

To prove Theorem 5, we use a method similar to the one used by Dvoretzky and Motzkin [9] to prove the original ballot theorem.

We start with some preliminary definitions. A lattice path can be expressed as a sequence that consists of n elements with value 1 and k elements with value $-\frac{n}{k}$, where the elements with value 1 correspond to the vertical segments of the path and the elements with value $-\frac{n}{k}$ correspond to its horizontal segments. If the given lattice path is x-strict, then the corresponding sequence starts with 1, and between any two elements of the form $-\frac{n}{k}$ there is at least one 1. Let us call any such sequence an *x-strict sequence*. Clearly, there is a bijection between x-strict lattice paths and x-strict sequences.

Note that the sum of all elements of an x-strict sequence is 0. An x-strict sequence is called *good* if every partial sum from the starting element of the sequence is positive, and *bad* otherwise. A *circular arrangement* is a clockwise arrangement of elements of a sequence on a circle. We say that two circular arrangements are *equivalent* if one can be obtained from the other by a circular shift. For example, in Figure 2 an x-strict lattice path from (0,0) to (3,4) corresponds to a circular arrangement. The segment p^0p^1 corresponds to the 12 o'clock element in the circular arrangement. The x-strict sequence $1, -\frac{4}{3}, 1, -\frac{4}{3}, 1, 1, -\frac{4}{3}$ is bad because the partial sum $1 + (-\frac{4}{3})$ of the first two elements is not positive. A *circular set for an x-strict sequence* is the set of all x-strict sequences with the same circular arrangement. We partition the set $\mathcal{A}$ into disjoint circular sets $C_1, C_2, \cdots, C_t$. We will now prove that the proportion of good sequences in each circular set is at most $\frac{1}{n}$.

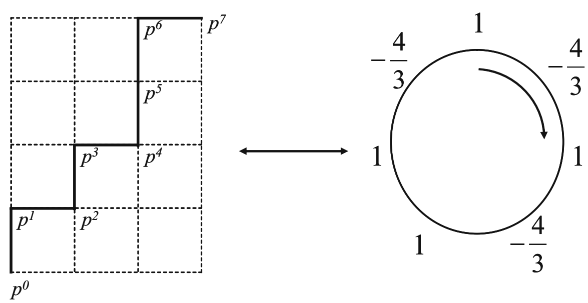


Fig. 2. An x-strict lattice path from (0,0) to (3,4) and the corresponding circular arrangement

Lemma 3. *If there exists a good sequence in a circular set for an x-strict sequence, then it is unique.*

Proof. Let $\mathtt{s}$ be a good sequence $\mathtt{s}_1, \mathtt{s}_2, \cdots, \mathtt{s}_{n+k}$ in a circular set. Let $\mathtt{s}'$ be another good sequence $\mathtt{s}_i, \mathtt{s}_{i+1}, \cdots, \mathtt{s}_{n+k-1}, \mathtt{s}_{n+k}, \mathtt{s}_1, \cdots, \mathtt{s}_{i-1}$ in the same circular set starting at $\mathtt{s}_i$. Since $\mathtt{s}$ is a good sequence, we have $\sum_{1 \leq j \leq i-1} \mathtt{s}_j > 0$. Therefore, we have $\sum_{i \leq j \leq n+k} \mathtt{s}_j < 0$, a contradiction with $\mathtt{s}'$ being a good sequence. □

Lemma 4. *The circular set of an x-strict sequence contains at most n distinct x-strict sequences. Moreover, if this circular set contains a good sequence, then it contains exactly n distinct x-strict sequences.*

Proof. Any x-strict sequence must start with 1 and there are n 1's in the circular arrangement for an x-strict sequence. Hence, there are at most n x-strict sequences in a circular set. Moreover, some of these n sequences may coincide. However, if the circular set contains a good sequence, then by Lemma 3 it is

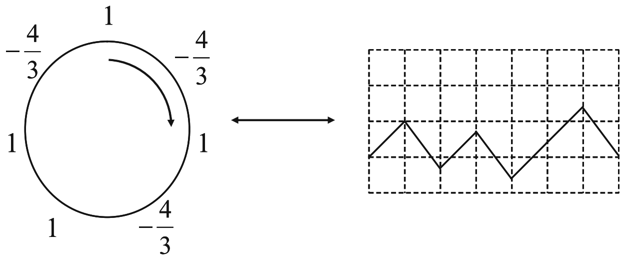


Fig. 3. A circular arrangement and the corresponding path

unique. Denote this good sequence by $\mathtt{a} = \mathtt{a}_1, \mathtt{a}_2, \ldots, \mathtt{a}_{n+k}$. Since $\mathtt{a}_1$ is uniquely determined from the circular arrangement, all the possible n x-strict sequences for this circular arrangement are distinct. □

We will refer to a path from $(0,0)$ to $(n+k,0)$ that has north-east steps of slope 1 and south-east steps of slope $-\frac{n}{k}$ as a *diagonal path*. There is a natural bijection between circular arrangements and diagonal paths (see Figure 3). We say that two diagonal paths are *equivalent* if the corresponding circular arrangements are equivalent. Consider the diagonal path in Figure 3. It is easy to see that the equivalent diagonal path that starts from its lowest point corresponds to the unique x-strict sequence in Lemma 3.

Lemma 5. *If n and k are relatively prime, then the circular set of any x-strict sequence contains exactly one good sequence.*

Proof. The circular set of an x-strict sequence $\mathtt{a} = \mathtt{a}_1, \mathtt{a}_2, \cdots, \mathtt{a}_{n+k}$ can be represented as a diagonal path (see Figure 3). Since n and k are relatively prime, the diagonal path has a unique minimum point. Suppose that this point corresponds to a point $\mathtt{a}_i$ in the circular arrangement. Then the sequence $\mathtt{a}_i, \mathtt{a}_{i+1}, \cdots, \mathtt{a}_{i+n+k-1}$ is good, and is in the same circular set as the sequence $\mathtt{a}$. The uniqueness follows from Lemma 3. □

We now prove Theorem 5.

Proof. We partition the set $\mathcal{A}$ into disjoint circular sets $C_1, C_2, \ldots, C_t$. Then $|\mathcal{A}| = \sum_{1 \le m \le t} |C_m|$. Let $\mathcal{B}$ be the set of all x-strict lattice paths that do not touch the line $y = \frac{n}{k}x$. Such paths correspond to good sequences, so by Lemma 3 and Lemma 4, we have $|\mathcal{B}| \le \frac{1}{n} \sum_{1 \le m \le t} |C_m| \le \frac{1}{n}|\mathcal{A}|$. Since $|\mathcal{A}| = \binom{n}{k}$, we obtain $|\mathcal{B}| \le \frac{1}{n}\binom{n}{k} = \frac{1}{k}\binom{n-1}{k-1}$. The first statement is proved. If n and k are relatively prime, the above inequalities become equalities by Lemma 5. □

5 Upper Bound

By Lemma 1, we can assume that $\{v_1, \ldots, v_n\} = \{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$. Let $\mathcal{Y}_{k,n}$ be the collection of all subsets of $\{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\}$ of size k. Any set $Y \in \mathcal{Y}_{k,n}$ is of

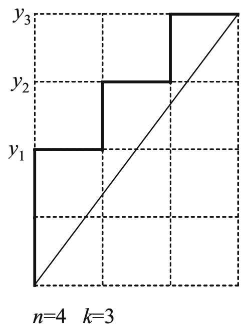


Fig. 4. Correspondence between an x-strict lattice path that does not touch the line $y = \frac{n}{k}x$ and $Y = \{\frac{1}{y_1}, \dots \frac{1}{y_k}\} \in \mathcal{Y}_{k,n}$, where $n = 4$ and $k = 3$

the form $\{\frac{1}{y_1}, \frac{1}{y_2}, \cdots, \frac{1}{y_k}\}$; we will assume $y_1 < y_2 < \cdots < y_k$. By Theorem 2, we have $\mathbf{ONOPT_n} = \sum_{k=1}^{n} E_T[g(T)] = \sum_{k=1}^{n} \sum_{y \in \mathcal{Y}_{k,n}} \frac{1}{\binom{n}{k}} \times \frac{1}{k} \max_{1 \le j \le k}(\frac{j}{y_j})$. Set $g(k,n) = |\{Y \in \mathcal{Y}_{k,n} \mid \frac{1}{y_k} = \frac{1}{n}, \frac{k}{n} > \frac{i}{y_i}$ for $1 \le i < k\}|$. The last condition in the definition of $g(k,n)$ corresponds to requiring that $y_i > i\frac{n}{k}$ for $1 \le i < k$. Any set Y that satisfies this condition corresponds bijectively to an x-strict lattice path that does not touch the line $y = \frac{n}{k}x$. Therefore, $g(k,n)$ is equal to the number of x-strict lattice paths that do not touch the line $y = \frac{n}{k}x$ (see Figure 4). By Theorem 5, we have $g(k,n) \le \frac{1}{k}\binom{n-1}{k-1}$. Thus, we have proved the following lemma.

Lemma 6. $g(k,n) \le \frac{1}{k}\binom{n-1}{k-1}$.

Let $y(k,n) = \sum_{Y \in \mathcal{Y}_{k,n}} max_j(\frac{j}{y_j})$. We will now prove an interesting inequality for $y(k,n)$. For ease of notation, set $\sum_{k+1 \le i \le n} \frac{1}{i} = 0$ for $k = n$.

Lemma 7. *We have* $y(k,n) \le \binom{n-1}{k-1} + \binom{n-1}{k-1} \sum_{k+1 \le i \le n} \frac{1}{i}$ *for all* $n \ge 1$ *and* $1 \le k \le n$.

Proof. We prove the lemma by using induction on k and n. The statement is clearly true for the base case: $y(1,n) = \sum_{1 \le i \le n} \frac{1}{i}$ for all $n \ge 1$.

Now suppose that $n \ge 2$ and $2 \le k \le n$. By the induction hypothesis, we can assume that the statement of the lemma holds for $y(k-1, n-1)$ and $y(k, n-1)$. To prove that it holds for $y(k,n)$, we need to consider two cases.

Case 1. $\frac{1}{y_k} \ne \frac{1}{n}$.
In this case, by the induction hypothesis, we have

$$\begin{aligned} \sum_{Y \in \mathcal{Y}_{k,n} \text{ s.t. } \frac{1}{y_k} \ne \frac{1}{n}} max_j(\frac{j}{y_j}) &= y(k, n-1) \\ &\le \binom{n-2}{k-1} + \binom{n-2}{k-1} \sum_{k+1 \le i \le n-1} \frac{1}{i}. \end{aligned} \quad (1)$$

Case 2. $\frac{1}{y_k} = \frac{1}{n}$.
Let I_Y be indicator variable given by $I_Y = 1$ if $\max_{1\le j\le k}(\frac{j}{y_j}) = \frac{k}{n}$ and $I_Y = 0$ otherwise. We get the following inequalities.

$$\begin{aligned}
&\sum_{Y\in\mathcal{Y}_{k,n} \text{ s.t. } \frac{1}{y_k}=\frac{1}{n}} max_j(\frac{j}{y_j}) \\
&\le \sum_{Y\in\mathcal{Y}_{k,n} \text{ s.t. } \frac{1}{y_k}=\frac{1}{n}} \left(\max_{1\le j\le k-1}(\frac{j}{y_j}) + I_Y(\frac{k}{n} - \frac{k-1}{n-1})\right) \\
&\le y(k-1,n-1) + g(k,n)(\frac{k}{n} - \frac{k-1}{n-1}) \qquad (2)\\
&\le \binom{n-2}{k-2} + \binom{n-2}{k-2}(\sum_{k\le i\le n-1}\frac{1}{i}) + g(k,n)(\frac{k}{n} - \frac{k-1}{n-1}) \\
&\le \binom{n-2}{k-2} + \binom{n-2}{k-2}(\sum_{k\le i\le n-1}\frac{1}{i}) + \frac{\binom{n-1}{k-1}}{k}(\frac{k}{n} - \frac{k-1}{n-1}).
\end{aligned}$$

The first inequality comes from the following fact. If $\frac{k}{y_k}$ is the largest element in $\{\frac{1}{y_1}, \frac{2}{y_2}, \cdots, \frac{k}{y_k}\}$, then the second largest element of this set is at least $\frac{k-1}{n-1}$. Therefore, $\max_{1\le j\le k}(\frac{j}{y_j}) \le \max_{1\le j\le k-1}(\frac{j}{y_j}) + I_Y(\frac{k}{n} - \frac{k-1}{n-1})$. The second inequality comes from the definition of $g(k,n)$. The third inequality follows from the induction hypothesis. The fourth inequality comes from Theorem 5. If we combine the inequalities (1) and (2), we get the following desired result.

$$\begin{aligned}
y(k,n) &\le \binom{n-2}{k-1} + \binom{n-2}{k-1}(\sum_{k+1\le i\le n-1}\frac{1}{i}) + \binom{n-2}{k-2} \\
&\quad + \binom{n-2}{k-2}(\sum_{k\le i\le n-1}\frac{1}{i}) + \frac{\binom{n-1}{k-1}}{k}(\frac{k}{n} - \frac{k-1}{n-1}) \\
&= \binom{n-1}{k-1} + \binom{n-1}{k-1}(\sum_{k+1\le i\le n}\frac{1}{i}).
\end{aligned}$$

□

We can now give an upper bound of $\mathbf{B}_\infty$.

Theorem 6. $\mathbf{B}_\infty \le 2$.

Proof.

$$\begin{aligned}
\mathbf{ONOPT_n} &= \sum_{k=1}^{n}\sum_{Y\in\mathcal{Y}_{k,n}} \frac{1}{\binom{n}{k}} \times \frac{1}{k} max_j(\frac{j}{y_j}) \\
&\le \sum_{k=1}^{n} \frac{1}{k\binom{n}{k}} \left(\binom{n-1}{k-1} + \binom{n-1}{k-1}(\sum_{k+1\le i\le n}\frac{1}{i})\right)
\end{aligned}$$

$$= \sum_{k=1}^{n} \frac{1}{n} \left(1 + \sum_{k+1 \le i \le n} \frac{1}{i} \right)$$
$$= 1 + \frac{1}{n} \left(\sum_{2 \le i \le n} \frac{i-1}{i} \right)$$
$$= 1 + \frac{n-1}{n} - \frac{1}{n} \left(\sum_{2 \le i \le n} \frac{1}{i} \right)$$
$$\le 1 + \frac{n-1}{n} - \frac{\ln(n+1) - \ln(2)}{n}.$$

The first inequality follows from Lemma 7. The last inequality follows from the simple fact that $\sum_{2 \le i \le n} \frac{1}{i} \ge \ln(n+1) - \ln(2)$.

Now, by Theorem 1, $\mathbf{B_n} \le 1 + \frac{n-1}{n} - \frac{\ln(n+1)-\ln(2)}{n}$ for any $n \ge 1$. Therefore, we conclude that $\mathbf{B}_\infty \le 2$. □

6 Concluding Remarks

In this paper, we improved previous results of Immorlica et al. for the balloon popping problem. We increased lower bound for the balloon popping problem from 1.64 to 1.659 by giving the algorithm **Bunch**. We can further increase the lower bound by the following method: Let $y = 2$ and $z = 2$ until the $\frac{1}{6}$-balloon is processed. After that, let $z = 3$. According to a Maple calculation, this gives an expected profit of at least 1.67.

For the online balloon popping mechanism, we decreased upper bound from 4.33 to 2 by extending the ballot theorem. Since a simulation result by Immorlica et al. shows that the upper bound may be greater than 1.89, we think that the upper bound in this paper is almost tight for online balloon popping mechanisms. Getting exact bound for the online balloon popping problem is open. Another open problem is finding the optimal solution for the balloon popping problem.

Acknowledgments. We would like to thank anonymous reviewers including Bernhard von Stengel for valuable comments and suggestions. We give special thanks to Edith Elkind for her pure contribution to improve presentation and exposition in this paper.

References

1. Abrams., Z.: Revenue maximization when bidders have budgets. In: Proceedings of SODA 2006, pp. 1074–1082 (2006)
2. Andelman, N., Mansour, Y.: Auctions with budget constraints. In: Hagerup, T., Katajainen, J. (eds.) SWAT 2004. LNCS, vol. 3111, pp. 26–38. Springer, Heidelberg (2004)

3. Balcan, M.F., Blum, A., Mansour, Y.: Item pricing for revenue maximization. In: Proceedings of EC 2008, pp. 50–59 (2008)
4. Bar-Yossef, Z., Hildrum, K., Wu, F.: Incentive-compatible online auctions for digital goods. In: Proceedings of SODA 2002, pp. 964–970 (2002)
5. Blum, A., Hartline, J.: Near-optimal online auctions. In: Proceedings of SODA 2005, pp. 1156–1163 (2005)
6. Borgs, C., Chayes, J., Immorlica, N., Mahdian, M., Saberi, A.: Multi-unit auctions with budget-constrained bidders. In: Proceedings of EC 2005, pp. 44–51 (2005)
7. Dobzinski, S.: Two randomized mechanisms for combinatorial auctions. In: Proceedings of APPROX-RANDOM 2007, pp. 89–103 (2007)
8. Dobzinski, S., Nisan, N., Schapira, M.: Truthful randomized mechanisms for combinatorial auctions. In: Proceedings of STOC 2006, pp. 644–652 (2006)
9. Dvoretzky, A., Motzkin, T.: A problem of arrangements. Duke Math. J. 14, 305–313 (1947)
10. Feige, U.: On maximizing welfare when utility functions are subadditive. In: Proceedings of STOC 2006, pp. 41–50 (2006)
11. Goulden, I.P., Serrano, L.G.: Maintaining the spirit of the reflection principle when the boundary has arbitrary integer slope. J. Comb. Theory Ser. A 104(2), 317–326 (2003)
12. Guruswami, V., Hartline, J.D., Karlin, A.R., Kempe, D., Kenyon, C., McSherry, F.: On profit-maximizing envy-free pricing. In: Proceedings of SODA 2005, pp. 1164–1173 (2005)
13. Immorlica, N., Karlin, A.R., Mahdian, M., Talwar, K.: Balloon popping with applications to ascending auctions. In: Proceedings of FOCS 2007, pp. 104–112 (2007)
14. Mahdian, M., Saberi, A.: Multi-unit auctions with unknown supply. In: Proceedings of EC 2006, pp. 243–249 (2006)
15. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized on-line matching. In: Proceedings of FOCS 2005, pp. 264–273 (2005)
16. Mirrokni, V., Schapira, M., Vondrak, J.: Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In: Proceedings of EC 2008, pp. 70–77 (2008)
17. Mohanty, S.G.: Lattice path counting and applications. Academic Press, London (1979)
18. Myerson, R.B.: Optimal auction design. Mathematics of Operations Research 6, 58–73 (1981)

Anarchy, Stability, and Utopia: Creating Better Matchings

Elliot Anshelevich, Sanmay Das, and Yonatan Naamad

Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
{eanshel,sanmay,naamay2}@cs.rpi.edu

Abstract. We consider the loss in social welfare caused by individual rationality in matching scenarios. We give both theoretical and experimental results comparing stable matchings with socially optimal ones, as well as studying the convergence of various natural algorithms to stable matchings. Our main goal is to design mechanisms that incentivize agents to participate in matchings that are socially desirable. We show that theoretically, the loss in social welfare caused by strategic behavior can be substantial. However, under some natural distributions of utilities, we show experimentally that stable matchings attain close to the optimal social welfare. Furthermore, for certain graph structures, simple greedy algorithms for partner-switching (some without convergence guarantees) converge to stability remarkably quickly in expectation. Even when stable matchings are significantly socially suboptimal, slight changes in incentives can provide good solutions. We derive conditions for the existence of approximately stable matchings that are also close to socially optimal, which demonstrates that adding small switching costs can make socially optimal matchings stable.

1 Introduction

This paper investigates the social quality of stable matchings. The theory of stable matching has received attention because of its many applications, including matching graduating medical students to residency programs [1], and matching kidney donors with recipients [2]. Most of the work on stable matching has assumed that the agents being matched have some preference ordering on who they would like to be matched with, without assigning a concrete utility for agent i being matched with agent j [3,4,5, inter alia]. This is natural, because stability as a concept does not need the stronger requirement of ascribing utilities to outcomes: it only needs the ranking of matchings from the perspective of every agent.

Matching problems, however, often bring with them outcomes that need to be evaluated in terms of cardinal utility. This occurs, for example, in pair programming, a central practice of the software engineering methodology known as Extreme Programming [6]. The utility of a matching is a function of the productivity of a pair of programmers working together. In kidney exchange, as well as many other stable matching scenarios, the goal is not only to form stable matchings, but also to form a matching with high overall utility.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 159–170, 2009.

The properties of matching mechanisms determine the utilities received by agents in these situations. A good mechanism for kidney exchange could make donors happier with their decision to donate while arranging the best possible matches for recipients. A good mechanism for pairing programmers would lead to the best possible programming productivity for their employer. Inevitably, there is a tradeoff between stable matchings, which are pairwise (or groupwise) rational, and socially optimal matchings (for our purposes, for the rest of this paper we assume simple additive social utilities, so that the socially optimal matching is the one that maximizes the sum of utilities received by each individual). The central question of mechanism design for matching markets is how to get people into "good" matchings, however "good" is defined. Almost all the work on matching mechanism design has focused on engineering stable matchings. This work has met with significant large-scale success in applications like matching graduating medical students to residency programs, and matching students to public high schools [7,1]. Some of this work, especially recent work on designing high school student matches, also explicitly seeks to realize the best matchings for one side of the market (in the high school case, the best matchings for students), but the notion of welfare is weak pareto-optimality among the set of stable matches for one side of the market [8].

Our focus is on extending our understanding of matching problems in situations where we are concerned with social welfare in terms of utility, instead of just stability and choice among stable outcomes. Several alternatives may be available in these situations, ranging from purely centralized allocation based on information available to a matchmaker, to purely individual decision-making based on personal preferences. The first set of questions that arises can be divided into three categories: (1) How bad are stable matchings when compared with socially optimal ones? (2) Can agents find stable matchings on their own? What are the outcomes of algorithms they may actually use in practice? (3) How can we incentivize agents to participate in matchings that are socially desirable?

Our Results. We initiate an investigation of the questions described above in the context of two-sided matchings, and give both theoretical and experimental results. Specifically, we study the effects of different network structures and utility distributions on the price of anarchy: the ratio of social utilities achieved by stable and optimal matchings respectively. We find that in most cases the stable matching attains close to the optimal social welfare (generally above 90%). We characterize some situations where the price of anarchy can be more substantial, and then study a potential means of incentivizing good stable matchings in Section 5. We consider *approximate stability*, which corresponds to the addition of a switching cost to the mechanism, so that an agent would have to pay in order to deviate from the current matching. We show both theoretically and experimentally that the addition of a small switching cost can greatly improve the quality of stable solutions. Finally, in Section 6 we consider several greedy algorithms for partner-switching, and show experimentally that they converge quickly to stability for some simple yet natural distributions of utilities, as well as prove convergence guarantees.

2 Matching, Stability, and Social Welfare

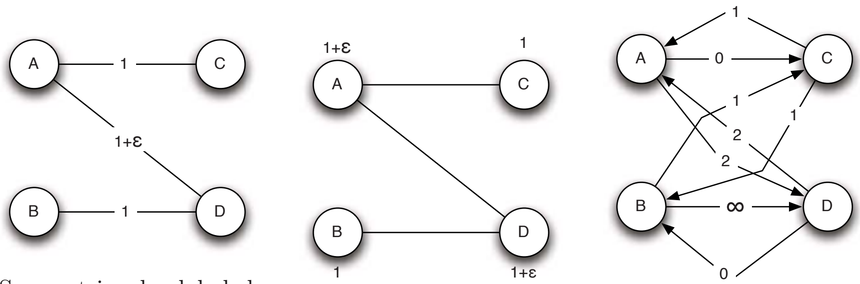
Matching, the process of agents forming beneficial partnerships, is a fundamental social process. Examples of matching with self-interested agents range from basic social activities (marriage, mate assignment [9]), to the core of economic activity (matching employees and employers [10]), to recent innovations in health care (matching kidney donors and recipients [11]). The process of matching can be complex, since (1) agents can have complicated preferences, and, (2) agents are self-interested: they care mostly about their own welfare, and would not obey a centralized matching algorithm unless it was to their benefit.

For this reason, the outcomes of matching processes are usually analyzed in terms of *stability*, the requirement that no collection of agents could form a group together, and become better off than they are currently [3]. For the classic "stable marriage" problem [12], this corresponds to the lack of desire of any pair to drop their current partners and instead match with each other. While stable matchings may be natural outcomes, desirable for various reasons, there are few guarantees on the quality and social welfare of stable matchings. Most research on matchings of self-interested agents has focused on (1) defining outcomes with stability as the goal, (2) computing stable outcomes and understanding their properties (ranging from the seminal work of Gale and Shapley [12] to algorithms that try and compute "optimal" matches, for example by minimizing the average preference ranking of matched partners [13]), and (3) designing truthful preference-revealing mechanisms (such as in public school matches [8]). Questions about the social welfare of stable matchings have been less studied.[1] There has been almost no research on constructing socially desirable stable outcomes, partly because in most situations one cannot instruct self-interested agents on what to do in order to engineer such outcomes, since an agent will only follow instructions if it benefits them personally.

An increasing body of literature in behavioral economics (e.g. [14]), however, suggests that desirable outcomes can be achieved by giving people a little "nudge" in certain directions, by altering their incentives slightly while still leaving them with freedom to choose their own actions. Small changes that greatly improve a social system are easy to identify in some situations: for example, making retirement savings 401(K) plans opt-out rather than opt-in increases participation dramatically. Finding similar changes in matching scenarios is more difficult because of the complexity of a system where any agent's actions can theoretically affect a large number of other agents.

Before addressing the mechanism design question of how to achieve better social outcomes, we first need to address the question of whether or not stable matching can lead to substantial social losses. For this question to make sense, we first need an objective function that measures the quality of a matching. One of the reasons why the social quality of stable matchings is usually not addressed

[1] One of the desiderata for matching students with schools or medical students with residencies can be to compute the stable matching that is best (typically) for the students, but this is a different notion of welfare.



Symmetric edge-labeled preferences

Vertex-labeled preferences

Asymmetric edge-labeled preferences

Fig. 1. Worst-case realizations of the price of anarchy. In each case the socially optimal matching is $\{(A, C), (B, D)\}$ but the only stable matching pairs A and D.

is because the agents in question are assumed to have a preference ordering on their possible partners, without a specific utility function that states how good a match would be.[2] Here we are specifically concerned with contexts where every agent has a utility function, not just a preference ordering: that is, for every possible partner v, an agent has a value $U(v)$ specifying how happy it would be to be matched with v.

The tradeoff between stable matchings and socially optimal matchings is quantified by the *price of anarchy*: the ratio between the maximum possible social utility and the utilities of equilibrium outcomes (stable matchings). Understanding the price of anarchy is important, since it acts as a bound on the amount of improvement in stable matchings that better mechanisms could yield.

Price of Anarchy Bounds. The price of anarchy can vary widely depending on the problem instance and the preference structure. Figure 1 illustrates some cases where the stable matching is highly socially suboptimal (discussed in more detail in the next section). In two of the underlying types of graph structures, the price of anarchy is at most two (and the bound can be tight), while in the third the social utility of the stable matching can be arbitrarily bad compared with the socially optimal one. But how bad are stable matchings in expectation? This question is tackled in detail in Section 4. Empirically, we find that despite the potentially bad worst-case behavior, across many different random distributions of preferences and several graph structures the price of anarchy tends to be lower.

Creating Better Stable Matchings. Given the agents' utilities, the social-welfare maximizing matching can be computed by finding a maximum weighted matching on a graph. We cannot just force people to accept such a matching because of individual preferences. But what if we could suggest a good matching, and provide some incentives for agents to go along with those matchings? This is

[2] Measures like average preference ranking [13] can be hard to justify. For example, agent A might greatly prefer its first choice to its second, while agent B only slightly prefers its first choice.

the subject of Section 5. We consider changing incentives to make more socially desirable matchings become stable by adding switching costs into the system. We show both theoretically and empirically that a small amount of incentives can greatly affect the quality of stable matchings.

Convergence to Stability. Another natural question we ask is whether stable matchings will arise in practical situations, where each participant does not want to submit his or her preferences to a centralized matchmaker. Previous work has focused especially on randomized best response dynamics [15,16]. We know that simple decentralized partner switching algorithms can fail to converge to stable matchings [15]. However, what happens in cases where the structure of preferences obeys some extra constraints? We explore this question in Section 6.

3 The Matching Model

We are concerned with pairwise matching problems. While we focus on bipartite graphs, (most of) our results also hold for general graphs, and in our experiments we did not find a significant difference between the quality of matchings in bipartite and non-bipartite graphs. We assume that each agent gains some utility from being paired up with another agent. The utility of remaining unmatched is assumed to be 0. We consider each agent as a vertex in a graph G, and only agents u and v with the edge (u, v) being present in G are allowed to match with each other. In two-sided matching scenarios, the agents can be separated into two types, one on each side of the graph, and no edges are allowed between agents of the same type.

We consider several different utility structures:

1. **Vertex-Labeled Graphs.** A vertex-labeled graph is defined as $G = (V, E, w)$ where V is the set of vertices, E is the set of (undirected) edges, and w is a vector of weights corresponding to the vertices. When two vertices u and v are in a matching, the agent corresponding to u receives utility $w(v)$ and the agent corresponding to v receives utility $w(u)$. These graphs correspond to a situation where being paired with agent X will yield the same utility to any agent Y allowed to match with X, independent of the identity of Y.
2. **Symmetric Edge-Labeled Graphs.** A symmetric edge-labeled graph $G = (V, E, w)$ is different in that the weights w correspond to edges rather than vertices. When two vertices u and v are in a matching, the agents corresponding to both u and v receive utility $w(\{u, v\})$. These graphs reflect situations where the utility received by both members of a pair is the same, perhaps determined by their combined output when working together – for example, pair programming may be judged by the productivity of the pair. Markets with these types of utilities are called "correlated two-sided markets" in [15].
3. **Asymmetric Edge-Labeled Graphs.** An asymmetric edge-labeled graph $G = (V, E, w)$ is the same except that edges are now directed, and the utility received by agent u in a matching that includes the pair u, v is given by $w(u, v)$, while the utility received by v is given by $w(v, u)$. This is the

most general case, in which each agent receives an unconstrained value from each agent they may possibly be paired with.

We also consider combinations of the above models, such as when agent u's utility for being matched with v has a vertex-labeled component $w(v)$, as well as an edge-labeled component $w(u, v)$. The types of utilities mentioned above arise in many contexts including market sharing games [17] and distributed caching games [18]. In the context of marriage markets, vertex-labeled graphs are equivalent to what Das and Kamenica call *sex-wide homogeneity of preferences*, and edge-labeled graphs are equivalent to what they call *pairwise homogeneity of preferences* [19].

4 The Price of Anarchy

In general, the price of anarchy is the ratio between the social utility of the (worst) equilibrium outcome of a game and the maximum social utility possible in that game. The usual definition relates the largest social welfare achievable to the social welfare of the worst Nash equilibrium. In the context of matching, we move from the concept of Nash equilibrium to the concept of stable equilibrium described above, because stable outcomes are determined by the possibility of pairwise deviations rather than individual deviations.

The price of anarchy can vary widely depending on the problem instance and the preference structure. Figure 1 illustrates some cases where the stable matching is highly socially suboptimal (the price of anarchy is high) in the three different preference settings for two-sided matching described in Section 3. Below we present price of anarchy bounds for the three models we consider.

Observation 1. *In symmetric edge-labeled graphs, the social utility of any stable matching is at least one-half of the social utility of the optimum matching.*

In other words, the price of anarchy is at most 2. The socially optimal matching is simply the maximum-weight matching in this model. The above observation is a special case of Theorem 1 (see Section 5), but it can also be seen to follow from two facts: (1) Any stable matching can be returned by an algorithm that examines edges greedily by magnitude, adding them to the matching if the vertices involved have not yet been matched (the particular stable matching produced depends on the procedure for breaking ties between equal-weighted edges), and (2) Any greedy solution to the maximum weighted matching problem is within a factor of two of the optimal solution. This argument holds generally, even for non-bipartite graphs. Figure 1(a) provides an example of a graph where this bound is achieved, showing that the bound is tight.

Observation 2. *In vertex labeled graphs the social utility of any stable matching is at least one-half of the social utility of the optimum matching.*

This is a consequence of Theorem 2 (see Section 5 for further discussion). Again, Figure 1(b) provides an example of a graph where this bound is achieved.

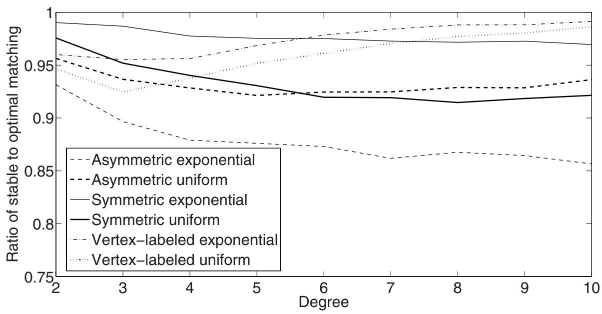


Fig. 2. Average ratio of the realized stable matching to the maximum weighted matching in three different preference models when utilities are sampled at random from exponential and uniform distributions with the same mean (0.5: the rate parameter is 2 for the exponential and the support of the uniform is $[0, 1]$). Reported values are averaged over 200 runs. There are 100 agents on each side of the matching market in all cases. The X axis shows the degree of each node. Note that the ratio is very high, almost never dropping below 85%, even in individual runs.

Observation 3. *In asymmetric edge-labeled graphs, the social utility of the stable matching can be arbitrarily bad compared with the socially optimal matching.*

Consider the case in Figure 1(c) – the utility received by agent B from being matched with Agent D is arbitrarily high, but the pair is not part of the stable matching, so the loss in utility can be unbounded. Again this argument holds for non-bipartite graphs as well.

These are worst-case constructions. A natural question is what the price of anarchy is like in realistic graphs with different distributions over utilities. We examine several different distributions of utilities within the three models described above, and also consider different graph structures in order to get a sense of the potential practical implications of these price of anarchy results. We generate random graphs of the different types described above, with randomly sampled utilities, and compute both the maximum-weighted stable matching (the socially optimal matching) and a stable matching using the Gale-Shapley algorithm (in all cases considered here, except one described in more detail below, the proposing side does not affect the outcome in expectation because preference distributions are symmetric).

Figure 2 shows that when utilities are randomly distributed according to two common distributions (exponential and uniform, although this result seems to be robust across many different distributions), the social loss due to stability is not particularly high in any of the three models we describe above. This is not surprising for vertex labeled graphs – since any person in the matching will contribute the same to the total utility regardless of whom they are matched with (for example, every perfect matching is socially optimal). As the average degree of each vertex increases, the number of agents getting matched increases, and the ratio quickly reaches 1, because all stable matchings become perfect at some point. However, the result is considerably more surprising for the other two cases, particularly for asymmetric edge-labeled preferences. The only case in which the ratio goes below 0.9 is for exponentially distributed utilities with asymmetric edge-labeled preferences (the ratio stops declining significantly beyond degree 10). For asymmetric edge labeled graphs, it makes sense that the ratio declines as the degree of the graph gets larger, because it becomes possible to construct matchings that are socially much better. Our experiments show that the value of the optimal matching grows quickly (since it has more options available), while the value of stable matching grows slowly (since it is hampered by the stability constraint). The actual high percentage is quite surprising given that in theory, the ratio could be arbitrarily bad. The uniform distribution ratios are generally higher than those for the exponential distribution because the uniform distribution enforces a compression in the range of high utilities by capping utilities at 1.

Some additional empirical results are presented in the full version of this paper [20]. They show that the results above are not particular to random bipartite graphs, but also hold for a variety of common networks, like preferential attachment networks and small-world networks. "Unbalancing" the network by making one side's range of utilities significantly higher than the other's can lead to a higher price of anarchy. Finally, it is worth noting that the price of anarchy is not the only important measure – for example, we show in the [20] that increasing the heterogeneity of tastes can lead to a higher price of anarchy, but increased utility for everyone.

5 Improving Social Outcomes

In this section, we consider how to improve the quality of stable matchings. We consider the addition of a switching cost to the mechanism so that an agent would have to pay in order to deviate from the current matching. We find that it is possible to improve the quality of social outcomes substantially by making only small changes to the incentives of the agents, and thus without drastically changing the nature of the matching market. An *approximate equilibrium* is a solution where no agent gains more than a small factor in utility by deviating. In the case of matching, we consider the following notion of approximately-stable matching.

Definition 1. *A matching is called* α-stable *if there does not exist a pair of agents not matched with each other who would both increase their utility by a factor of more than* α *by switching to each other.*

If $\alpha = 1$, then this is exactly a stable matching. An α-stable matching also corresponds to a stable solution if we assume that switching has a cost. In other words, in the presence of switching costs, the set of stable matchings is simply the set of α-stable matchings without switching costs.

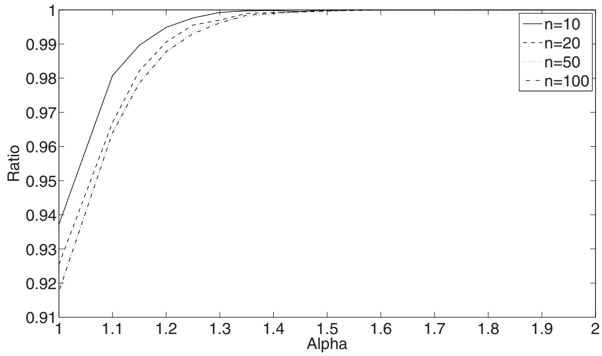


Fig. 3. Ratio of the social utilities of best α-stable and socially optimal matchings as a function of α when the matchings are constructed according to our algorithm in symmetric edge-labeled graphs. The dramatic increase between $\alpha = 1$ and $\alpha = 1.1$ shows that introducing even small switching costs has the potential to produce significant social benefits. Preferences were sampled uniformly at random on $[0, 1]$.

How does increasing α improve the quality of stable matchings? We are specifically concerned with the *price of stability* [21], which is the ratio of the utility of the *best* stable matching relative to the optimum matching. Much recent work in network design and routing [22,23] has considered the price of stability in various contexts. It is especially important from the point of view of a mechanism designer with limited power, since it can compute the best stable solution and suggest it to the agents, who would implement this solution since it is stable. Therefore, the price of stability captures the problem of optimization subject to the stability constraint.

Below we present various theoretical bounds, showing that for symmetric edge-labeled graphs, there always exists an α-stable matching with utility of at least $\frac{\alpha}{2}$OPT (where OPT is the value of the optimum matching), and that in vertex-labeled graphs, there always exists an α-stable matching with utility at least $\frac{\alpha}{1+\alpha}$OPT. We provide a constructive algorithm for finding such an α-stable matching. This shows that by increasing α, we can implement much better stable solutions than for $\alpha = 1$, and decrease the price of stability. Empirical results using this algorithm show an even more dramatic improvement than predicted by the theoretical bounds. Figure 3 shows that for $\alpha = 1.1$ we already obtain a tremendous improvement in the quality of stable matching, essentially obtaining stable matchings that are as good as a matching with maximum utility. This means that adding a switching cost as small as five or ten percent can make an enormous difference in the quality of stable matchings. In many situations, it is reasonable to believe that a central controller can compute a good α-stable matching, assign agents to that matching, and only allow them to deviate on payment of the switching cost.

For edge-labeled graphs, in the presence of switching costs of a factor α, the price of anarchy is at most 2α, but the price of stability is at most $2/\alpha$. This means that as we increase α, there begin to be stable matchings that are worse, but there always exists a stable matching that is close to optimal. For $\alpha = 1$, these bounds coincide, giving us the result that all stable matchings are within a factor of 2 from the maximum weight matching. For $\alpha = 2$, this gives us the easily verifiable fact that the optimum matching is 2-stable.

Theorem 1. *Let OPT be the value of the socially optimal matching. In any undirected edge-labeled graph, there exists an α-stable matching whose social utility is at least $\frac{\alpha}{2}OPT$. Furthermore, the social utility of any α-stable matching is at least $\frac{1}{2\alpha}OPT$.*

Our proofs appear in the full version of this paper [20].

Similar results hold for vertex labeled graphs. The price of anarchy is at most $1+\alpha$ and the price of stability is at most $(1+\alpha)/\alpha$. For $\alpha = 1$ this gives us the observation in Section 4 (notice that while it is easy to show a correspondence between stable matchings for edge-labeled and vertex-labeled graphs, the same does not hold for α-stable matchings).

Theorem 2. *Let OPT be the value of the maximum-weight perfect matching. In any vertex-labeled graph, there exists an α-stable matching whose social utility is at least $\frac{\alpha}{1+\alpha}OPT$. Furthermore, the social utility of any α-stable matching is at least $\frac{1}{1+\alpha}OPT$.*

6 Convergence to Stability

While many good algorithms exist for computing stable matchings (Gale-Shapley being the most standard), we would like to consider more natural dynamics for forming stable matchings. Such dynamics are likely to occur in practice if there were no central planner to compute a matching for the agents, and if instead the agents tried to do what was best for themselves in a decentralized manner. In such cases, how likely is it that realistic algorithms yield stable outcomes?

We study the convergence properties of a particular decentralized partner-switching algorithm: first, sort the vertices randomly, then repeat until convergence: for each vertex, in the sorted order, find the best partner that vertex can be matched with. The vertex can be matched with a partner if an edge connects them and the deviation is utility-increasing for both the vertex and its new partner. The best partner is the one of these that yields maximum utility for this vertex. Add this new pair to the matching, removing any pairs that this vertex or its new partner were previously connected to.

This algorithm captures the intuitive notion that, in a society of agents, pairs take turns deviating from the current matching if it is in their interest to do so. We call each iteration through all agents a *phase*. Instead of iterating through all the agents in a fixed order, we could instead pick random agents to deviate at every step, as in [15]. None of our results change significantly in this case.

Theorem 3. *This algorithm converges to a stable matching after at most n phases in vertex-labeled and symmetric edge-labeled graphs.*

The simple decentralized algorithm described above converges to a stable matching in time $O(n^2)$, since each phase takes linear time. Notice, however, that if instead of switching to its best partner, the agents simply switched to a random improving partner, the same argument would guarantee convergence to a stable matching in an expected time of $O(n^2 d)$, where d is the maximum degree of the graph. In practice (see figure in [20]), on random utility distributions, the convergence time for vertex-labeled graphs does indeed appear to be quadratic, but the convergence time for symmetric edge-labeled graphs seems linear. We conjecture that the algorithm converges in expected linear time for these graphs, perhaps because good edges for one node are in expectation also good for the other node in the edge, because of the symmetric preferences.

Asymmetric Edge-Labeled. While Theorem 3 guarantees convergence for vertex-labeled and symmetric edge-labeled utilities, but in asymmetric edge-labeled graphs there are easy examples where this algorithm can cycle. In our experiments, however, for small n (the number of nodes on each side) the algorithm converged to a stable matching on all but a small percentage of cases, showing that the bad scenarios are not "typical." As n gets larger, this algorithm converges more and more rarely (approximately 2% less for every additional node), with convergence essentially non-existent for $n = 70$.

7 Discussion

This paper explores the prices of anarchy and of stability in matching markets. We demonstrate that even though the price of anarchy can theoretically be high, when utilities are randomly sampled, the loss in social welfare from strategic behavior is limited. This result encompasses many different graph and preference structures, and is experimentally robust. While the downside is limited, even this downside can be alleviated: a significant improvement in social welfare can be obtained by suggesting a good matching and requiring agents to pay small switching costs to deviate. We show this theoretically using an algorithm for constructing *approximately stable* matchings, and then demonstrate that the algorithm is effective in experiments. We also show that simple greedy partner switching algorithms can converge quickly to stable matchings in some graph structures. From a practical perspective, future work should include understanding real-world utility distributions and how they affect the social outcomes of matching as compared to random distributions of utilities. From a mechanism design perspective, it would be interesting to explore whether agents would choose to participate in a switching-cost based, designer-suggested matching mechanism.

References

1. Roth, A.E., Peranson, E.: The redesign of the matching market for American physicians: Some engineering aspects of economic design. American Economic Review 89(4), 748–780 (1999)
2. Roth, A.E., Sonmez, T., Ünver, M.U.: Pairwise kidney exchange. Journal of Economic Theory 125(2), 151–188 (2005)

3. Roth, A.E., Sotomayor, M.: Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Econometric Society Monograph Series. Cambridge University Press, Cambridge (1990)
4. Roth, A.E., Xing, X.: Jumping the gun: Imperfections and institutions related to the timing of market transactions. The American Economic Review 84(4) (1994)
5. Immorlica, N., Mahdian, M.: Marriage, Honesty, and Stability. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (2005)
6. Dawande, M., Kumar, S., Mookerjee, V., Sriskandarajah, C.: Maximum Commonality Problems: Applications and Analysis. Management Science 54(1) (2008)
7. Abdulkadiroglu, A., Pathak, P., Roth, A.: The New York City High School Match. American Economic Review 95(2), 364–367 (2005)
8. Abdulkadiroglu, A., Pathak, P., Roth, A.: Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match. American Economic Review (to appear, 2009)
9. Becker, G.: A Treatise On The Family. Family Process 22(1), 127–127 (1983)
10. Jovanovic, B.: Job Matching and the Theory of Turnover. The Journal of Political Economy 87(5), 972 (1979)
11. Abraham, D., Blum, A., Sandholm, T.: Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In: Proc. of ACM Conf. on Electronic Commerce, pp. 295–304. ACM Press, New York (2007)
12. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly 69(1), 9–15 (1962)
13. Irving, R., Leather, P., Gusfield, D.: An efficient algorithm for the "optimal" stable marriage. Journal of the ACM (JACM) 34(3), 532–543 (1987)
14. Thaler, R., Sunstein, C.: Nudge. Yale University Press, New Haven (2008)
15. Ackermann, H., Goldberg, P., Mirrokni, V., Roglin, H., Vocking, B.: Uncoordinated two-sided markets. In: Proceedings of the 9th ACM Conference on Electronic Commerce, EC (2008)
16. Roth, A., Vande Vate, J.: Random Paths to Stability in Two-Sided Matching. Econometrica 58(6), 1475–1480 (1990)
17. Goemans, M., Li, L., Mirrokni, V., Thottan, M.: Market sharing games applied to content distribution in ad hoc networks. IEEE Journal on Selected Areas in Communications 24(5), 1020–1033 (2006)
18. Mirrokni, V.: Approximation Algorithms for Distributed and Selfish Agents. PhD thesis, Massachusetts Institute Of Technology (2005)
19. Das, S., Kamenica, E.: Two-sided bandits and the dating market. In: Proc. IJCAI, Edinburgh, UK, August 2005, pp. 947–952 (2005)
20. Anshelevich, E., Das, S., Naamad, Y.: Anarchy, Stability, and Utopia: Creating Better Matchings, http://www.cs.rpi.edu/~eanshel
21. Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, E., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. In: Proc. FOCS, pp. 295–304 (2004)
22. Anshelevich, E., Dasgupta, A., Tardos, E., Wexler, T.: Near-optimal network design with selfish agents. In: Proceedings STOC, pp. 511–520. ACM Press, New York (2003)
23. Christodoulou, G., Koutsoupias, E.: On the Price of Anarchy and Stability of Correlated Equilibria of Linear Congestion Games. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 59–70. Springer, Heidelberg (2005)

Equilibria in Dynamic Selfish Routing

Elliot Anshelevich[1] and Satish Ukkusuri[2]

[1] Dept. of Computer Science, Rensselaer Polytechnic Institute, Troy, NY
eanshel@cs.rpi.edu
[2] Dept. of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, Troy, NY
ukkuss@rpi.edu

Abstract. In both transportation and communication networks we are faced with "selfish flows", where every agent sending flow over the network desires to get it to its destination as soon as possible. Such flows have been well studied in time-invariant networks in the last few years. A key observation that must be taken into account in defining and studying selfish flow, however, is that a flow can take a non-negligible amount of time to travel across the network from the source to destination, and that network states like traffic load and congestion can vary during this period. Such flows are called dynamic flows (a.k.a. flows over time). This variation in network state as the flow progresses through the network results in the fundamentally different and significantly more complex nature of dynamic flow equilibria, as compared to those defined in static network settings.

In this paper, we study equilibria in dynamic flows, and prove various bounds about their quality, as well as give algorithms on how to compute them. In general, we show that unlike in static flows, Nash equilibria may not exist, and the price of anarchy can be extremely high. If the system obeys FIFO (first-in first-out), however, we show the existence and how to compute an equilibrium for all single-source single-sink networks. In addition, we prove a set of much stronger results about price of anarchy and stability in the case where the delay on an edge is flow-independent.

1 Introduction

The concepts of "selfish flow" and "selfish routing" in time-invariant networks have been thoroughly explored over the last few years (see, for example, [18, 19, 22]). These models apply to networks that involve routing by a large number of independent self-interested agents, such as transportation networks, Internet routing, and peer-to-peer file sharing systems. In all these systems, individual agents using the network (vehicles on highways, Internet peer-to-peer clients, etc.) can be expected to be somewhat "selfish", and may only be interested in optimizing their own performance metric when making routing decisions. Understanding equilibria in such networks is crucial in order to measure system efficiency and performance, and to design suitable mechanisms that improve the properties of the system.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 171–182, 2009.

Unlike in the usual selfish routing models, however, in many networked systems the state is a function of both the number of agents and *the time when the agents use the network.* In addition, a flow (or equivalently, the agents/entities that constitute the flow) can *take a non-negligible amount of time to travel across the network from the source to destination, and network state like traffic load and congestion can vary during this period.* For instance, in hurricane evacuations, the number of people evacuating in the transportation network is a time-dependent process which can vary considerably during the time it takes to move from homes to safe shelters. In the context of the Internet, network congestion levels (and thereby, download speeds) can vary significantly within a few hours, which is the typical time-scale for downloads of large video files in peer-to-peer systems.

We study routing done by self-interested agents in the context of such flows, which we call "dynamic flow" (or equivalently, "flow over time"). The behavior of dynamic flows is very different from static (time-invariant) flows, as was seen before in numerous papers studying dynamic flows from a centralized point of view [14,10,12] (i.e, without self-interested agents). To further illustrate the difference between static and dynamic flow, consider that in static selfish flow, the congestion of a link is a function of all the users that use this link, aggregated over time. This is problematic, however, in time varying flows. In our model, when carefully choosing their best route, user i does not consider the congestion on link e based on the total number of users that traverse e, or even on the users that are currently using e, but instead considers the congestion on e that will take place when i reaches e. For a detailed description of our model, see Section 2.

In this paper we study equilibria in dynamic flows, and prove various bounds about their quality, as well as give algorithms on how to compute them. We concentrate on Nash equilibria in dynamic flow models, and show that their properties can be quite different from the ones in static models. We call such equilibria "dynamic equilibria" to differentiate them from Nash equilibria in static selfish flow. The notion of dynamic Nash equilibrium in this context refers to a system state where each agent is not better off by deviating from its chosen solution (which can consist of the agent's route, start and waiting times, etc.). As this paper illustrates, the variation in network state as the flow progresses through the network results in the fundamentally different and significantly more complex nature of dynamic flow equilibria problems, as compared to those defined in static network settings.

Background and Related Work. While both dynamic flows (a.k.a. flows over time) and selfish flows have been studied extensively, few outside the transportation community [15] have yet attempted to study flows that are *selfish* and *take time to traverse a link.* Below we outline some of the related work on these topics.

Static Selfish Flows. Traditional computer science research concerning routing of self-interested agents in networks has mostly focused on *static* flows [18,4]. In a static routing game, we are given a graph $G = (V, E)$ of links, with latency functions $\ell_e(x)$ for each edge e (these functions are usually assumed to be non-decreasing and convex). We are also given a source s and sink t (or possibly many sources and sinks), along with some amount of flow that desires to be routed from s to t. This flow is sometimes considered to be *non-atomic* (composed of an infinite number of users, each controlling an infinitesimal amount of flow) or *atomic* (composed of a finite number of users, each controlling a discrete amount of flow). In this paper, we focus on non-atomic flow, although many of our results can be adopted to the atomic case. Each user is able to choose the route along which its flow proceeds, and since the users are self-interested, they will choose the route that has minimum congestion. Specifically, if x_e is the total amount of flow on edge e, then the congestion or delay on path P is defined to be $\sum_{e \in P} \ell_e(x_e)$. When considering the solution quality for such flows, the most common measure is the total congestion (which also corresponds to the social welfare), measured by $\sum_{e \in E} x_e \ell_e(x_e)$. The centralized optimal solution is considered to be the flow that minimizes this value (and therefore maximizes social welfare). Many properties are known about such static selfish flows, such as results about the prices of anarchy [18,5] and stability [6], and the fact that a Nash equilibrium exists and (for certain classes of games) is unique.

Flows over time. The single major difference between the unsolved problems we address and most traditional network flow research is the notion of time passing while flow (i.e., cars/packets) move through the network. This type of flow is often called "flow over time" or "dynamic flow", and is applicable not only to all kinds of transportation networks, but also a wide range of data communication networks. Specifically, a flow over time means that flow on different links in the network can change over time, and more importantly, a flow requires a certain amount of *transit time* to travel through each link, this amount possibly dependent on the current congestion. Unlike in static flow systems, the amount of flow or congestion on a link changes at every time step, as some flow enters and some flow leaves this link, adding an extra temporal component. There has been some excellent research in flows over time using various optimization techniques (for surveys, see [12,2,16,17]). The work in [13,9] is especially relevant to ours, since these papers consider traffic delays that are flow-dependent, as we do in Sections 3 and 4. All of this research, however, does not take into account the strategic nature of agents in the system. One of the main goals of this paper is to combine the techniques used for analyzing dynamic flows with the ones used for static selfish flows [7].

Dynamic Equilibrium in Transportation Networks. Due to the inherent nature of time-varying flow in transportation networks, dynamic flow (referred to as *Dynamic Traffic Assignment*) models are heavily used in transportation network planning, operations, and evaluating real-time systems. These are typically classified into two categories: analytical models based on mathematical programming formulations, and simulation-based heuristic models. Extensive work has been performed for both types of approaches and an overview of this literature can be found in [21,15]. A key limitation of the previously developed models is that even under simplified assumptions, the models lack rigorous theoretical results on the existence, uniqueness, and algorithms to compute dynamic flow equilibria. While simulation models allow us to compute a "solution", it is difficult to guarantee that it is indeed a dynamic equilibrium. Simply put, our work provides a much needed theoretical foundation to the dynamic strategic flow concept, while building on the work done by transportation researchers. Portions of the results in this paper were presented at DTA 2008 (International Symposium on Dynamic Traffic Assignment), and the full version of this paper can be found at [1].

2 Model and Our Results

In this section we describe our general model of strategic flow over time. In our model we are given a road or computer network, represented by a directed graph $G = (V, E)$. We also have some flow demands, with source-sink pairs $\{s_i, t_i\}$ such that a unit of flow desires to move from s_i to t_i. This flow is selfish and non-atomic, so a player in this case corresponds to an infinitesimal amount of flow, and the goal of every player is simply to reach its destination in the least amount of time possible. The strategies of the players consist of picking a path from their source to their destination.

Everything we defined so far is exactly the same as in the usual selfish routing model. The main difference is the congestion function. Every link has a function $d_e(x, H_e^t)$, which determines how long it takes x units of flow that enter edge e at time t to traverse this edge. This amount can depend not just on the amount of flow x, but also on the "history" of the edge usage. Specifically, define H_e^t to be the set of amounts of flow using e before time t, and how long ago it entered e before time t. For example, if 2 units of flow entered e one time step before t, and 1/3 units of flow entered e two timesteps before t, then $H_e^t = \{(2, 1), (1/3, 2)\}$. This is an extremely general way to model flow over time. The fact that the delay d_e depends on the history vector H_e means that, for example, the delay for x flow entering at time t can depend on the total flow that is currently on the edge, or on the time until all previous flow leaves the edge, etc. We assume that the functions d_e are monotone increasing, i.e., that increasing the amount of flow in H_e or in x can only increase delays. Given these congestion functions and the behavior of other users, each user chooses a path from s_i to t_i that would minimize its delay.

What makes our model drastically different from static selfish flow is that it takes *time* for a user to traverse a link. As we show in Section 4, the properties of our model are also very different from the static flow model. For example, equilibria may not exist in dynamic flows, and unlike for static non-atomic flows, they may not be unique. This results partially from the fact that unlike static flows, self-interested dynamic flows do not form a congestion game [4], and modeling dynamic flow games simply as repeated static flow games can result in great inaccuracies.

Flow-Dependent Delays Obeying FIFO. We are especially interested in dynamic flows where the FIFO (first-in first-out) property is obeyed. This is certainly true for most communication networks, where packets are forwarded according to their arrival order, and is a common assumption for transportation networks as well, since FIFO is largely enforced in time of high congestion, and is enforced on average always. Because of this, in Section 3 we consider the general model described above, but with the assumption that for all e,

$$d_e(x, H_e^t) + t \geq d_e(x', H_e^{t'}) + t' \tag{1}$$

where x enters e at time t, and x' enters e at time $t' < t$ (so that $(x', t-t') \in H_e^t$). This condition simply states that if flow x' enters an edge before x, then it also leaves before x. For a discussion of some concrete models that fit into this framework, see Section 3.1.

In Section 4 we show that for delay functions satisfying FIFO, it is still possible that no equilibria exist, and that the price of anarchy is unbounded. On the positive side, however, we show in Section 3 that for single-source single-sink networks obeying FIFO, a Nash equilibrium always exists, and can be computed efficiently.

Flow-Independent Delays. In Section 5 we focus on a very special case of flow independent delay functions. In this model, the travel time on a link $e \in E$ does not depend on the amount of flow on that link, so the delay d_e is constant. Concurrently with our work, [11] showed many interesting properties of a continuous version of the model with flow-independent delays. While the flow-dependent delay model above has extremely different properties from static flow, in the case of flow-independent delays there is a close relationship between static and dynamic flows [14].

We exploit this relationship to give price of stability bounds for dynamic flows with flow-independent delays. Specifically, we show that there always exists a Nash equilibrium that is as good as the centralized optimum. We also consider the case when the players' cost functions do not simply correspond to travel time. For example, a player's cost might be a function of both travel time and the congestion on the links taken. We show that if these cost functions are linear, then we can compute an equilibrium that is at most 4/3 more expensive than the centralized optimum (and in general it will be α more expensive, where α is a factor dependent on the class of cost functions in [19, 18]).

3 Flow-Dependent Delays Obeying First-In First-Out (FIFO)

An important goal in modeling dynamic flows is to understand the desirable properties of a good dynamic equilibrium model. In this section we assume that the delays on the links satisfy the first-in first-out (FIFO) property. The FIFO property can be defined as that any unit of flow A entering a link e before some flow unit B also exits the link before B; in other words overtaking is not allowed[1]. It is easy to show that simple inflow [9] or exit flow models violate the FIFO condition. An inflow model is a model where the travel time for x flow entering a link at time t

[1] For a precise definition, see Inequality 1.

depends only on the size of flow x. In other words, the delay function is of the form $d_e(x)$, and does not rely on H_e. To see that inflow models violate FIFO, consider the case when there is a sharp increase in traffic flow at time t followed by a rapid decrease in flow at time $t+1$. Then FIFO will be violated since the time taken for a flow entering at time t is much higher than the flow entering at time $t+1$. Several authors in the transportation community have tried to overcome this limitation by assuming that the number of vehicles on a link at time t is a function of both inflow and exit flow [8,3]. While these models work well for most practical traffic flows, FIFO can still be violated in such models.

In this section we assume that all delay functions satisfy the FIFO property, i.e., that Inequality 1 always holds. There is a range of models which satisfy this condition, including point queue models and other models that provide approximate "positions" (based on microscopic traffic simulations) of the flow currently on the link (i.e., how far along the link it has traveled so far) or/and a function of speed of the flow. We describe some of these models in Section 3.1.

Even if the FIFO property holds, dynamic equilibria can be extremely different from static equilibria. As we show in Section 4, in multi-source multi-sink networks, equilibria may not exist, and the price of anarchy can be unbounded. This is in contrast with static flows, where an (essentially unique) equilibrium always exists, and can be computed efficiently [18].

For single-source single-sink networks, however, dynamic flows behave much better in the presence of FIFO. Below we assume that there is a single unit of flow starting from a node s and with destination t. If more than a single unit of flow is present, we can always scale the flow so that there is only a single unit: the important thing is that flow only leaves s at one moment in time, since otherwise this essentially becomes a multi-source problem. For the single-source single-sink case, we prove that there exists a Nash equilibrium for this general class of dynamic models, and provide an algorithm to compute the equilibrium solution. To the best of our knowledge, this is the first theoretical result on the existence/computation of Nash equilibria for this general class of models. The proofs of all our results can be found in the full version [1].

Theorem 1. *If delay functions obey FIFO, when a single (splitable) unit of flow desires to get from a source s to a sink t, a Nash equilibrium always exists and can be computed efficiently.*

The proof of this theorem essentially shows that equilibria in our model correspond exactly to static equilibria in the appropriate static network. Since static equilibria are efficiently computable (especially for linear congestion functions), this theorem provides us with an algorithm for computing dynamic equilibria as well. In addition to the existence of Nash equilibria, we can also show that dynamic equilibria are unique in the presence of strict FIFO.

Proposition 2. *With the same assumptions as in Theorem 1, if* strict *FIFO is satisfied (i.e., Inequality 1 is satisfied without equality), then the dynamic Nash equilibrium is unique.*

3.1 Concrete Models Satisfying FIFO

There are many realistic flow models that satisfy the FIFO property. Unfortunately, the delay functions d_e for most of the models cannot be written in closed form and do not possess a general analytical solution. However, these models can be solved using numerical methods using various discrete time forward or backward differencing methods for solving first-order differential equations. To illustrate the types of functions d_e that obey FIFO, we present a specific model below. For another, simpler set of functions, see the examples in Section 4.

Variations of the Point Queue Model. Consider the following form of what is known in the transportation community as the point queue model. Suppose that a flow x enters the tail of a link e at time t, and assume that if there were no flow on the link at time t, then x will traverse the link in a fixed travel time which can be a constant c_e or, to more accurately account for congestion effects, an expression such as $c_e x$. Now suppose instead that some flow y were already present on the link at time t. In this model, we will think of flow x as traveling on edge e with constant velocity V_x, picked so that it never passes any flow that is in front of it, which is also traveling at constant velocity. Specifically, let $\tau(y)$ be the time at which the last unit of the y flow exits the link e. If y entered e at time t', then it is traveling at velocity $V_y = 1/(\tau(y) - t')$. Since it is now time t, the flow y has traversed $\frac{t-t'}{\tau(y)-t'}$ of the edge. If we now set $V_x = 1/(c_e x + \tau(y))$, then the flow x will never overtake any flow in front of it, guaranteeing the FIFO property. To see this, notice that

$$\frac{\tau - t'}{\tau(y) - t'} > \frac{\tau - t}{c_e x + \tau(y)}$$

for all $\tau \geq t$. This is true because $\tau c_e x + \tau t' + t\tau(y) > t' c_e x + tt' + t'\tau(y)$, which holds because $\tau \geq t > t'$. Since $\frac{\tau - t}{c_e x + \tau(y)}$ is exactly the position of the x flow at time τ if it is traveling at velocity V_x, then the flow x always lags behind flow y. The travel time for flow x in this case will be $d_e(x, H_e^t) = c_e x + \tau(y)$.

In fact, any travel times that result in the x flow exiting the edge after $\tau(y)$ would satisfy FIFO. For example, we could instead say that x exits the edge at time $\max(c_e x + t, \tau(y))$. This would correspond to the x flow traveling along the link at a top speed of $1/c_e x$, but being unable to pass any flow that is in front of it. In general, any model where flow x has a top speed for traveling on an edge, with its actual speed dependent on the presence or on the density of the flow ahead of x, fits into our framework. FIFO is satisfied in such a model as long as the speed constraints are such that flow never passes anyone ahead of it.

4 Lower Bounds and Examples with No Equilibrium

In the previous section we saw that for the single-source single-sink case of dynamic flow satisfying FIFO, a unique Nash equilibrium exists, and can be computed efficiently. Unfortunately, as we show below, this equilibrium can be much worse than the optimal solution, leading to large prices of anarchy and stability (which are the same in this case, since the equilibrium is unique). We also give an example with multiple sources where an equilibrium does not exist at all, and an example with multiple equilibria.

Example with Unbounded Price of Anarchy. In the proof of Theorem 1, we established that the equilibria of our dynamic model correspond to static equilibria in the same graph G with cost functions $g_e(x)$. In this case $g_e(x)$ is the time it would take x amount of flow to traverse edge e if there were no flow ahead of it on e. The price of anarchy in our model can be very different from the price of anarchy for static flows, however, since while the cost of the equilibrium remains the same, the optimal solution in the dynamic model can be very different from the optimal solution for static flow. This leads to the cost of the optimal solution being very small compared to the cost of the equilibrium.

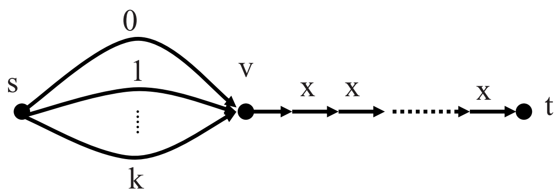


Fig. 1. An example with unbounded Price of Anarchy. The edges are labeled with values $d_e(x, \emptyset)$.

To illustrate this, consider the example in Figure 1, where we have k parallel paths leading from s to v, followed by a path of 100 edges leading from v to t. Suppose the parallel paths have constant delay, with the i'th path having delay $i-1$. That is, if e is the i'th parallel edge, then set $d_e(x, H_e^t) = i-1$. The 100 edges in the bottleneck path each have a delay such that if x amount of flow enters an edge, with no other flow currently on this edge, then it will take x time units for this flow to leave the edge. Any such delay functions that obey FIFO will give us the desired example. To be fully concrete, however, we will give a specific example of d_e, defined recursively. First, we set $d_e(x, \emptyset) = x$. In general, if the last amount of flow that entered e before t was an amount x' at time t', then the latest element of H_e^t is $(x', t-t')$. In such a case, we define $d_e(x, H_e^t) = \max\{d_e(x', H_e^{t'}) + t' - t, 0\} + x$. These functions satisfy the desired condition, since if there is no other flow on edge e at time t, then $d_e(x', H_e^{t'}) + t' \leq t$, and so $d_e(x, H_e^t) = x$. It is also easy to show that these functions satisfy FIFO (see [1]).

For Multi-source, Nash Equilibria May not Exist. We have shown above that the price of anarchy can be very high in most FIFO models, since the optimal dynamic solution can "stagger" the flow by breaking it up into small pieces, while an equilibrium must keep all the flow together so all of it arrives at the same time. For the multi-source version of our FIFO model, the situation is even worse, since a Nash equilibrium may not exist at all. While below we present a multi-source multi-sink example with no equilibrium, there also exist such examples with only a single sink node (but multiple source nodes). We give one such example in [1].

Consider the example in Figure 2, with functions $d_e(x, \emptyset)$ as shown in the figure. We define the functions $d_e(x, H_e^t)$ as in the previous example, so that FIFO is satisfied, and so that if x units of flow enters e at time t, and no flow is present on e at that time, then it takes $d_e(x, \emptyset)$ time for this x flow to traverse the link. In the previous example, this value $d_e(x, \emptyset)$ was equal to x, but now it can vary between edges as specified in the figure. For example, for the edge from v to t_2, if x flow enters this edge, with no other flow currently on this edge, then it will take $100x$ time units for this flow to leave the edge. If an edge is not labeled with a function, then we say that the delay on this link is always 0.

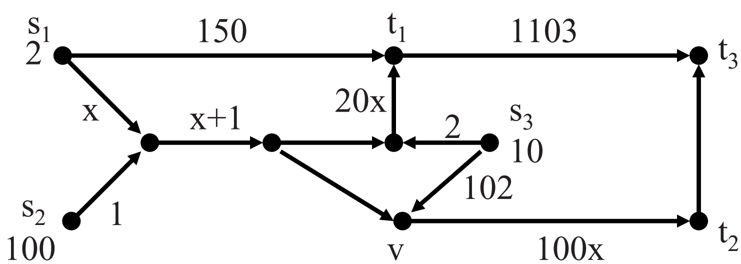


Fig. 2. A multi-source multi-sink example with no Nash Equilibrium. The edges are labeled with values $d_e(x, \emptyset)$.

There are three demands d_1, d_2, d_3 with sources (s_i, t_i) for $i = 1, 2, 3$, and with flow sizes of 2, 100, and 10 respectively. All these demands leave their sources at the same time.

Proposition 3. *The example pictured in Figure 2 has no pure dynamic Nash equilibrium.*

Equilibrium is Not Unique. As we proved in Proposition 2, if strict FIFO is obeyed, then there exists only a unique dynamic equilibrium. In the full version of this paper [1], we present a simple example where equilibria are not unique when FIFO is violated.

5 Flow-Independent Delays

In this section, we consider an important special case of *Flow-Independent* delay functions. Specifically, in the Flow-Independent model, we assume that each link e has a delay d_e, and any flow that enters this link at time τ leaves this link at time $\tau + d_e$. In other words, the delay function is simply $d_e(x, H_e^t) = d_e$, a constant. This is flow-independent in the sense that the delay d_e does not depend on the amount of flow on the link. If edges are uncapacitated, then the best thing for any flow (whether selfish or not) would be to proceed on the fastest path from the source to the sink. Therefore, we assume that each edge also has a capacity c_e so that at every timestep, at most c_e units of flow are allowed to enter e.

In Section 4, we showed that dynamic flow models can behave very differently from static flow. For the flow independent case, however, there is a powerful relationship between static and dynamic flows. As several papers including [14] pointed out before, looking at a flow over time in the Flow Independent Model is the same as looking at a static flow in the appropriate time-expanded graph. Specifically, form a new graph G' with a set of nodes V^τ in G' for every timestep τ, and add edges $(v^\tau, w^{\tau+d_e})$ with capacity c_e to G' if and only if there is an edge $e = (v, w)$ in G. For every sink node t in G, we also form a special node t^* in G', and add edges (t^τ, t^*) for every time τ. It is easy to see that any static flow in G' corresponds exactly to a flow over time in G (even with multiple sources and sinks). In fact, [14] gave a PTAS for finding the centralized optimal solution in this model. Notice that in our model we do not allow "waiting" (a.k.a. "intermediate storage"), where flow stays at a vertex instead of immediately leaving it. See [1] for further discussion on this topic.

To understand the price of anarchy in the Flow Independent Model, we first consider the quality of Nash equilibria in the static flow model with capacities. It is easy to see that there can exist many equilibria in the presence of capacities, and in fact the price of anarchy can be unbounded [6], since the worst equilibrium can be much more expensive than the centralized optimum. This has led to the investigation of the *price of stability*. Recall that the price of stability is the ratio between the *best* Nash equilibrium and the centralized optimum. The best equilibrium can naturally be viewed as the optimum solution subject to the constraint that the solution be *stable*, with no agent having an incentive to unilaterally deviate from it once it is implemented. For models with a unique equilibrium, the price of stability coincides with the price of anarchy, but in the case of capacitated networks, [6] demonstrated that these can be dramatically different.

Theorem 4. *For Flow-Independent delays, the price of stability is 1. In other words, there exists a Nash equilibrium as good as the centralized optimum.*

Because of the correspondence between static flows in G' and dynamic flows in G, we know that comparing dynamic equilibria with OPT in G is the same as comparing static equilibria with OPT in G'. This tells us that the price of anarchy in the Flow-Independent Model is unbounded, using the results of [6].

Using this intuition, we can also generalize Theorem 4 as follows. Suppose that the disutility of a player traveling on path P is not simply its delay $\sum_{e \in P} d_e$ (as we assume in the rest of this paper), but is instead a function of how congested its route is. Specifically, for every edge e of G, suppose that there is some function $\delta_e^t(f_e^t)$ that shows the cost to a player using edge e at time t, with this cost dependent on the amount of traffic f_e^t entering edge e at time t. The total cost to a player using path P is then the sum of these costs, and minimizing the total cost to all the players is the same as maximizing social welfare. In other words, we assume that while it still takes a constant amount of time d_e for everyone to traverse edge e (independent of the amount of traffic f_e^t), the disutility $\delta_e^t(f_e^t)$ of a player using edge e actually increases as the amount of traffic increases, modeling the fact that people do not like to drive in heavy traffic, even if it does not cause them to be delayed. This can also model other modes of transportation (such as subway systems), where the travel time may not change with the number of users, but the utility of a user changes greatly (e.g., because the subway car is crowded, or there is no place left to sit).

These new player utilities do not change the possible flows, so the transformation between G and G' is still valid, as well as the correspondence between dynamic flows in G and static flows in G'. If we set the cost of an edge e^t in G' to be a function $\delta_e^t(f_e^t)$ instead of just a constant d_e, then the cost of a flow F' in G' is still the same as the total cost to all players in the corresponding dynamic flow F of G. This means that instead of thinking about dynamic flows in G, we can now think about static flows in G', and compare the Nash equilibria in G' with OPT. Using existing results about static flows [20], we know that in capacitated graphs, the price of anarchy can be very high, while the price of stability is at most $\alpha(A)$, a value depending on the class of possible latency functions A. For example, if the functions δ_e^t are linear in the amount of traffic, then $\alpha(A)$ equals $4/3$, and for polynomials of degree d and positive coefficients, $\alpha(A) = d/\log d$. In addition, we know that in graphs without capacities, the price of anarchy is at most $\alpha(A)$ [18], and so all equilibria are good compared to the optimal centrally planned solution.

Theorem 5. *The prices of anarchy and of stability for the Flow-Independent model with player cost functions δ_e are at most those for the static flow model with cost functions δ_e^t. For example, if δ_e^t are all linear functions, then:*

– *The price of stability is at most $4/3$ for capacitated graphs.*
– *The price of anarchy is at most $4/3$ for uncapacitated graphs.*

If instead of linear, the functions δ_e^t were polynomials with degree at most d, then the factor of $4/3$ above can be replaced with $d/\log d$.

Acknowledgements

We thank Lisa Fleischer and Neil Olver for illuminating discussions on modeling flows over time.

References

1. Anshelevich, E., Ukkusuri, S.: Equilibria in Dynamic Selfish Routing, http://www.cs.rpi.edu/~eanshel
2. Aronson, J.E.: A survey of dynamic network flows. Annals of Operations Research 20, 1–66 (1989)
3. Carey, M.: Link travel times ii: properties derived from traffic-flow models. Networks and Spatial Economics 4(4), 379–402 (2004)

4. Christodoulou, G., Koutsoupias, E.: On the price of anarchy and stability of correlated equilibria of linear congestion games. In: ACM Conference on Electronic Commerce, ESA (2005)
5. Cominetti, R., Correa, J.R., Stier-Moses, N.: Network games with atomic players. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 525–536. Springer, Heidelberg (2006)
6. Correa, J.R., Schulz, A.S., Stier-Moses, N.E.: Selfish routing in capacitated networks. Mathematics of Operations Research 29(4), 961–976 (2004)
7. Farzad, B., Olver, N., Vetta, A.: A priority-based model of routing. Chicago Journal of Theoretical Computer Science, Article #1 (2008)
8. Friesz, T., Luque, J., Tobin, R.L., Wie, B.W.: Dynamic network traffic assignment considered as a continuous time optimal control problem. Operations Research 37, 893–901 (1989)
9. Hall, A., Langkau, K., Skutella, M.: An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times. Algorithmica 47(3), 299–321 (2007)
10. Hoppe, B., Tardos, É.: The Quickest Transshipment Problem. Mathematics of Operations Research 25(1), 36–62 (2000)
11. Koch, R., Skutella, M.: Nash Equilibria and the Price of Anarchy for Flows over Time. Matheon Preprint 524 (2008)
12. Köhler, E., Möhring, R.H., Skutella, M.: Traffic Networks and Flows Over Time. In: Kramer, J. (ed.) DFG Research Center: Mathematics for Key Technologies. Berliner Mathematische Gesellschaft, pp. 49–70 (2002)
13. Köhler, E., Skutella, M.: Flows over time with load-dependent transit times. SIAM Journal on Optimization 15(4), 1185–1202 (2005)
14. Fleischer, L., Skutella, M.: Minumum cost flows over time without intermediate storage. In: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 66–75 (2003)
15. Peeta, S., Ziliaskopoulos, A.: Foundations of dynamic traffic assignment: The past, the present, and the future. Networks and Spatial Economics 1(3/4), 233–266 (2001)
16. Powell, W.B., Jaille, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhause, G.L. (eds.) Network Routing. Handbooks in Operations Research and Management Science, ch. 3, vol. 8, pp. 141–295. North-Holland, Amsterdam (1995)
17. Ran, B., Boyce, D.E.: Modeling Dynamic Transportation Networks. Springer, Berlin (1996)
18. Roughgarden, T.: Selfish Routing and the Price of Anarchy. MIT Press, Cambridge (2005)
19. Roughgarden, T., Tardos, É.: How bad is selfish routing? Journal of the ACM 49(2), 236–259 (2002)
20. Stier-Moses, N.: Selfish versus Coordinated Routing in Network Games. PhD Dissertation, Sloan School of Business, MIT (2004)
21. Waller, S.T., Ukkusuri, S.: A linear program for the user optimal dynamic traffic assignment problem. In: Review, Transportation Science (2003)
22. Wardrop, J.G.: Some Theoretical Aspects of Road Traffic Research. In: Proceedings of the Institute of Civil Engineers, Part II, vol. 1, pp. 325–378 (1952)

Stochastic Stability in Internet Router Congestion Games

Christine Chung[1] and Evangelia Pyrga[2]

[1] Department of Computer Science, University of Pittsburgh, PA
chung@cs.pitt.edu
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany
pyrga@mpi-inf.mpg.de

Abstract. Congestion control at bottleneck routers on the internet is a long standing problem. Many policies have been proposed for effective ways to drop packets from the queues of these routers so that network endpoints will be inclined to share router capacity fairly and minimize the overflow of packets trying to enter the queues. We study just how effective some of these queuing policies are when each network endpoint is a self-interested player with no information about the other players' actions or preferences. By employing the adaptive learning model of evolutionary game theory, we study policies such as Droptail, RED, and the greedy-flow-punishing policy proposed by Gao et al. [10] to find the stochastically stable states: the states of the system that will be reached in the long run.

1 Introduction

Ever since the first congestion control algorithms for TCP endpoints were introduced in [12], the important problem of congestion control at bottleneck routers on the Internet has garnered wide-spread attention. Several algorithms have been proposed for queue management and scheduling of packets in routers. Initially, such algorithms were designed under the assumption that all packets arriving at the routers come from TCP complying sources that produce packet flows with certain characteristics: all flows that become aware of congestion at the router (by seeing some of their packets dropped) will respond by reducing their transmission rates. However, TCP flows are not the only ones competing for available bandwidth or space in router queues. UDP flows behave in a completely different manner, tending to be more aggressive without sharing the same congestion control profile as TCP. Moreover, the assumption that future users will continue using the current TCP protocol seems questionable. Since there is no central authority governing their behavior, as users compete for bandwidth, they may very well change the way they respond to congestion.

Studying congestion control from a game theoretic perspective was therefore the natural next step. Using a variety of models, game theory has been used not only to find Nash equilibria (NE) when users are self-interested and routers employ existing methods, (e.g. FIFO with Droptail, or RED [8]) but also to design new router queuing policies, aimed at reaching good social outcomes in the presence of such users [10]. Such "good social outcomes" include the avoidance of congestion at routers, and thus avoidance of Internet congestion collapse, but also fairness of bandwidth sharing.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 183–195, 2009.

However, an approach commonly taken is to assume perfect information. Users are assumed to know the transmission rates of others and the congestion levels at the router, and use this information to compute a best response and optimize their utility. Even though such assumptions are standard, and even necessary, in the study of NE, they are not likely to be met in a setting like the Internet. Without such assumptions, can the equilibria be reached? Could there be a set of states, none of them necessarily a NE, such that the system gets essentially "trapped" cycling among the states in the set? These are the questions we are aiming to answer in this work.

Using a simple yet general model of the game played by internet endpoints at internet bottleneck routers, we provide the first (to our knowledge) analysis of this problem using stochastic stability, a classical solution concept from the adaptive learning model of evolutionary game theory. Evolutionary game theory's adaptive learning setting is suited especially well for the game of internet endpoints competing for bottleneck router capacity. In traditional game theoretic settings, each player must assume all other players are perfectly rational, and must be fully informed of each other's actions and preferences. When players are internet endpoints, such requirements seem unreasonable and quite unlikely to be met. In our evolutionary setting, under adaptive learning's imitation play, players need only know what you would expect them to know: what they themselves experience in each round of play. Then they use simple heuristics to decide, based on the results of their recent play, what strategy to employ for the next round. The simplicity of the model but also its ability to cope with limited information, make it particularly useful for modeling router congestion games.

To study our problem in this adaptive learning setting, we use a new model proposed by Efraimidis and Tsavlidis [7] called the *window game*. This model is not only simple, but more general than previous models in which players are usually assumed to be TCP endpoints with specific loss recovery properties. In the window game, the endpoints are modeled so that they can be thought of as using either TCP, UDP, or whatever transmission protocol they choose. There are n internet endpoints, each seeking to send an unlimited amount of traffic. But all endpoints encounter the same bottleneck router, which has capacity C. Each of the endpoints is a player that chooses a strategy: an integer-sized "window" between 0 and C. The window size can be thought of as the amount of the router's capacity being requested by the player, or the number of packets being sent by the player. The amount of capacity that the router actually allocates to each player is then determined by the router's queuing policy and the specified window sizes (capacity requests) of the other users. The utility of a player is defined as the number of successfully sent packets, minus the number of dropped packets times some factor $g \geq 0$. Hence, g represents the cost a player suffers by having one packet dropped.

We assume that this game is played repeatedly in rounds, in which every player chooses a strategy to play using imitation dynamics: sampling the outcomes of the rounds of play in its memory, and then imitating the strategy that served it best. However, with very small probability, each player fails to follow the imitation dynamics and chooses a strategy at random. Then, loosely speaking, the set of *stochastically stable states* represents the set of strategy profiles that have positive probability of being played in the long run, or, the states that the system eventually settles on. More details can be found in Section 2.1.

Our Results. The policies we deal with here have been studied with respect to Nash Equilibria before (see the Related Work section for more details), mainly though assuming that the rates at which the sources send their packets is described by a given rule, for instance, assuming Poisson rates. In our work, we make no such assumption, but employ the *window game* of Efraimidis and Tsavlidis [7]. We believe that this model is simple enough to allow interesting theoretical analysis, but still captures the essence of the game played between competing internet endpoints. We extend the results in [7] with respect to Nash Equilibria but also study the stochastic stability of the underlying games.

We begin by analyzing the two currently most well-known and widely-used router queuing policies: FIFO with Droptail and RED (Random Early Detection) [8]. When Droptail is used, all incoming packets are simply dropped once the queue is full. We show that for any reasonable value of g, the only NE and the only stochastically stable state is the state where all players send $\frac{g+1}{g}C\frac{n-1}{n^2}$ packets. This implies, for instance, that for a large number of flows and any value of $g \leq 1$ ($g = 1$ means that each player is hurt by each lost packet about the same as the amount they gain from each successful packet), the router is getting hit by roughly more than twice as much traffic as it has capacity. Next, we show that under RED queuing, in which the router starts dropping packets preemptively as soon as its buffer reaches a certain threshold $T < C$, things cannot get much better. For reasonable values of g, there is a single NE, which constitutes also the single stochastically stable state, in which the congestion at the router is still significant.

Finally we study a queue policy proposed by Gao et al. [10], in which any overflow is compensated for by dropping the packets belonging to the most demanding flow. This policy was designed, in a idealized setting, to have a unique NE such that the router capacity is equally shared among all flows and overflow is avoided. They also studied a non-idealized setting in which flows do not have perfect information, and all sources are restricted to fixed-rate Poisson rates except one, which can be arbitrarily aggressive. In this setting, they succeed at a more modest goal: the source that can be arbitrarily aggressive should not outperform the best Poisson source by much. In this work we show that this policy can actually do even better. We show that even if flows have no information about one another, and all of them can arbitrarily adjust their window sizes (so no flow is restricted to a fixed rate), the system will still converge to the fair equilibrium under adaptive learning with imitation dynamics.

Even though the stochastically stable states for the queuing policies we study turn out to coincide with the NE, what our results indicate is the following: even in the chaotic internet setting, where players have extremely limited information about the game and make instantaneous decisions, the NE will actually be reached.

Related Work. FIFO with Droptail is the traditional queue policy that has been employed widely in internet routers. As soon as the router queue is full, all subsequent incoming packets are dropped. For more information on Droptail and its variants, see [2]. RED [8] works similarly, but starts dropping packets with a certain probability as soon as the number of packets in the queue exceeds a threshold $T < C$. Both these policies punish all flows in a similar manner, regardless of whether they are "responsible" for causing the overflow or not. Specifically, the expected fraction of the demand of each flow that gets through the router is the same among all flows, those with moderate demands, and those with demands far exceeding their "fair share". The result is that flows with large demand can use more router capacity at the expense of lower-demand flows.

There have been methods suggested for inhibiting such behavior. The Fair Queueing algorithm [4] ensures the maxi-min fairness criterion: using round-robin for selecting the outgoing packets, every flow can at least obtain its "fair share." Even though this is a fair scheme, it comes at the cost of efficiency. It requires separate buffers for each queue and a lot of book-keeping, making it unusable in practice. A method that achieves the same result without the high computational cost at the routers was suggested in [19]. This method however, cannot be used independently in each router, as it depends on receiving flow-specific information from other routers.

CHOKe [17], on the other hand, is a stateless queue management scheme, which can be implemented in a router independently from what other routers use. When a packet arrives to the queue, it is compared to $M \geq 1$ packets chosen uniformly at random from those currently in the queue; if it comes from the same source as any of them, then both are dropped. There are both theoretical and experimental studies [20,15] suggesting its effectiveness at preventing greedy (e.g. UDP) flows from strangling moderate flows. However, as the number of greedy flows varies, the parameter M must also change in order to protect the more moderate flows from losing their fair share.

Gao et al [10] introduce a router queue management algorithm, which, unlike Fair Queuing, does not require separate buffers for each flow, but, under some assumptions, achieves the same (fair) NE as maxi-min fairness. The main idea is to keep track of the "greediest" flow. Whenever there is an overflow, the algorithm drops only packets that belong to this flow. [1] The *Prince* algorithm described in [7] works in a similar manner. The algorithm in [10] was aiming to fulfill, among others, the following two objectives. First, in an idealized environment of full information, the profile corresponding to maxi-min fairness is the unique NE. Second, removing the full information setup but restricting all flows but one to being Poisson sources of fixed rates, the unrestricted flow has no way of obtaining a throughput much better than that of the best Poisson flow.

There are several game theoretic results for congestion control. For a better introduction, we refer the reader to [18] and [14]. Akella et al. [1] study the equilibria of a TCP game, in which all flows use the Additive Increase Multiplicative Decrease (AIMD) algorithm. This is the method currently employed by TCP endpoints. The strategy sets consist of the possible values for the parameters of the algorithm. They show that even though the older TCP endpoint implementations can lead to efficient equilibria even with FIFO Droptail and RED router queue policies, this is no longer the case with newer implementations. They show that some measure of "network efficiency" can be established with a variant of CHOKe, assuming however that all flows are TCP. A lot of work has been devoted to game theoretic models in which all flows originate from Poisson sources and each source is allowed to vary the transmission rate [18,5,6]. The inefficiency of NE is studied, mainly in the case of a single bottleneck router, but also in

[1] Only in case that the overflow is greater than the number of packets of the greediest flow in the queue, will packets from other flows be dropped as well.

more general networks [11]. Kesselman et al. [14] consider a model in which the flows are explicitly deciding when to send new packets, instead of implicitly modifying their transmission rates.

An evolutionary game theoretic approach based on adaptive learning is used in [16] to analyze a game in which users set transmission rates for optimally receiving multimedia traffic. In [3], adaptive learning with imitation dynamics was used to analyze a load balancing game.

The Window Game model we study here was first proposed in [7], where it was used to find the NE in games between AIMD but also more general flows.

2 Model, Notation, and Background

To model internet endpoints competing for capacity at a bottleneck router, we use the window game of [7]. Let N be the set of players, $|N| = n$, with each player representing an internet endpoint. The strategy set for each player is the set of all possible window sizes, integer values between 0 and C, where C is the capacity of the bottleneck router. Let w_i be the window size requested by player i. Let $w = (w_1, w_2, \ldots, w_n)$; w is a *strategy profile* vector of the game. Let w_{-i} refer to the vector of all the strategies in w except w_i. Let $W = \sum_{i=1}^{n} w_i$ and let $W_{-i} = W - w_i$. The bottleneck router uses a (possibly randomized) queuing algorithm (like Droptail, RED, etc.), to decide how many of each player's packets to keep, and how many to drop. Therefore the queuing policy maps each strategy profile w to a corresponding vector that indicates for each player i how many of its w_i packets are kept (in expectation), $keep_i$, and how many are dropped, $w_i - keep_i$. As described in the previous section, $g \geq 0$ is a real value that indicates how much detriment a lost packet causes to each player. Then for any $i = 1 \ldots n$, function of i is $\pi_i(w) = (keep_i) - g(w_i - keep_i)$.

A *best response* to w_{-i} for each player i is then $br_i(w_{-i}) = \arg\max_{w_i} \pi_i(w_i, w_{-i})$.

2.1 Adaptive Learning and Imitation Dynamics

We now more formally present the relevant aspects of evolutionary game theory's adaptive learning model [9,21,22], as well as the imitation dynamics of [13]. A related, more detailed summary can be found in [3], in which adaptive learning and imitation dynamics are applied to a load balancing game.

In the adaptive learning model with imitation dynamics, each of n players has a finite memory of their own actions and payoffs in the previous m rounds of play. After each round, each player samples (uniformly at random) s of the m previous rounds of play, and then in the next round, plays the strategy (in our case, the window size) that yielded highest average payoff over the rounds that were sampled. In this way, the player is "imitating" the strategy that has served her well in the past.

These dynamics correspond to a Markov process P, where each state in the process is the history of the last m rounds of play. Each play history is comprised of m strategy profiles, and a state where all m strategy profiles are the same is called a *monomorphic* state.[2] The transition probabilities between states of the process are determined by the imitation dynamics described above. A *recurrent class* of a Markov process is a set of states such that there is zero probability of leaving the set once a state in the set has been reached, but positive probability of reaching any state in the set from any other state in the set. Josephson and Matros [13] prove the following about the process P.

Theorem 2.1 ([13]). *If $s/m \leq 1/2$, a subset of states is a recurrent class if and only if it is a singleton set containing a monomorphic state.*

If we now suppose that in each round, each player with probability $\epsilon > 0$ does not follow the imitation dynamics, but instead chooses a strategy at random, we have modified the Markov process so that there is always positive probability of eventually reaching any state from any other state. Therefore, there is a unique stationary distribution over the states in this modified process. We refer to this modified process as the *perturbed* Markov process, P^{ϵ} and the stationary distribution as μ^{ϵ}. The *stochastically stable states* (SSS) are those states h in this modified process for which $\lim_{\epsilon \to 0} \mu^{\epsilon}(h) > 0$.

A *better reply* is a unilateral strategy deviation by a player that gives that player at least as high a payoff as the original strategy profile. I.e., x is a better reply for player i if $\pi_i(x, w_{-i}) \geq \pi_i(w)$. A *cusber set* or a set "closed under single better replies," is a set of strategy profiles such that any sequence of better replies, by any sequence of players, starting from any strategy profile in the set, always leads to another strategy profile that is also in the set. A *minimal cusber set* is a cusber set such that if any strategy profile is removed, the remaining set is no longer a cusber set.

Theorem 2.2 ([13]). *Under imitation dynamics, the profiles in the set of stochastically stable states are a minimal cusber set or a union of minimal cusber sets.*

Note that the following corollary is an immediate consequence of Theorem 2.2.

Corollary 2.3. *If a single strategy profile comprises the only minimal cusber set in a game, then that is the only strategy profile in the set of stochastically stable states under imitation dynamics.*

For a more complete background on stochastic stability and imitation dynamics, we refer the reader to [22,13]. In what remains, we assume that $s/m \leq 1/2$.

3 Droptail

FIFO queues with Droptail are widely used in Internet routers. While the queue has not reached its capacity, incoming packets are inserted in the end of the queue. As soon as the capacity is reached, any new incoming packets are dropped. We will start by describing the window game model of Droptail, then discuss the NE, and finally prove that there is a single stochastically stable state that corresponds to the unique NE.

[2] For expository simplicity, if a monomorphic state has w as the strategy profile that fills its history, we will sometimes abuse notation and use w not just as the name of the strategy profile, but when the context is clear, as the name of the monomorphic state containing w.

Remember that for any profile w, we denote by W the total window size requested, i.e., $W = \sum_{i=1}^{N} w_i$. Under the Droptail routing policy, when $W > C$, the router chooses $W - C$ packets uniformly at random to be dropped. Therefore, for any player i with window size w_i, the expected number of packets of i that will enter the queue is $w_i \cdot C/W$, while $w_i \cdot (1 - C/W)$ will be dropped. Of course, when $W \leq C$ no packets will be dropped. This means that the expected payoff for player i can be expressed as

$$\pi_i(w) = \begin{cases} w_i & \text{if } w_i \leq C - W_{-i} \\ w_i \cdot \frac{C}{W} - g w_i \left(1 - \frac{C}{W}\right) & \text{if } w_i > C - W_{-i} \,. \end{cases} \tag{1}$$

We note that when the total window size equals the capacity, i.e., $w_i + W_{-i} = C$, then both pieces of the payoff function result in the same payoff. Therefore, for $W = C$ either of the two subcases can be used.

Definition 3.1. *Define d_g to be $\frac{g+1}{g}\, C\, \frac{n-1}{n^2}$.*

Efraimidis and Tsavlidis in [7] proved that, assuming $g \leq n-1$,[3] the profile $(d_g, ..., d_g)$ is the unique *symmetric* NE. In fact, as the next theorem states, that is the only NE for the case $g \leq n-1$. The proof, which involves first determining the best response function for each player, and then ruling out the possibility of all other NE, can be found in the full version of this paper.

Theorem 3.2. *If $g \leq n-1$, then the outcome in which each player's window size is $d_g = \frac{(g+1)C(n-1)}{n^2 g}$ is the only NE.*

In the following, we will assume that $g \leq n-1$, since the case where $g > n-1$ is of no practical relevance. We will now establish that the state $(d_g, \ldots, d_g)$ is the only SSS. Our proof uses the fact that any profile in a stochastically stable state is found in a minimal cusber set (Theorem 2.2), along with the fact that under Droptail the only minimal cusber set in our game is the NE profile itself. We first give two lemmas that allow us to establish the latter fact, by showing there is a better-reply path from any profile to the NE profile. Due to lack of space, we refer to the full version of this work for the proof of Lemma 3.3.

Lemma 3.3. *Let $w \neq (d_g \ldots, d_g)$, $W \geq C$. Within at most two better replies, a profile w' can be reached, such that for any k with $w_k = d_g$, $w'_k = d_g$, and there is some player i, such that $w_i \neq d_g$ and $w'_i = d_g$. Moreover, $W' \geq C$.*

Lemma 3.4. *For any $w \neq (d_g, \ldots, d_g)$, there is a finite sequence of better replies that leads to the profile $(d_g, \ldots, d_g)$.*

Proof. We note first that if $W < C$, then for any player i, playing $C - W_{-i}$ is a better response than w_i. Hence we will assume that $W \geq C$. Note that applying Lemma 3.3 to $w \neq (d_g, \ldots, d_g)$, we will obtain some w' such that still $W' \geq C$. Therefore, by simply invoking Lemma 3.3 at most n times, we can see that there is a path of (in total) at most $2n + 1$ better response moves from w to the profile $(d_g, \ldots, d_g)$. □

[3] Given that the number of flows that share the capacity of a bottleneck router is large, the case that $g > n-1$ is not realistic, and thus of no practical interest. For completeness, the NE for the case that $g > n-1$ are discussed in the full version of this work.

Theorem 3.5. *For $g \leq n-1$, the state in which every player plays d_g is the unique stochastically stable state.*

Proof. First of all, note that d_g is the unique better response to $W_{-i} = (n-1)d_g$. Therefore the profile $a = (d_g, \ldots, d_g)$ is a minimal cusber set. Moreover, by Lemma 3.4, there is a better response path from any $w \neq a$ to a. Therefore any other cusber set would have to contain a, which implies there is no other minimal cusber set. Hence, by Corollary 2.3, a is the only state that is stochastically stable. □

4 RED (Random Early Detection)

RED (Random Early Detection) [8] was meant to keep the average queue size low. It works similarly to Droptail, but starts dropping packets before the queue is full. When the total load at the router exceeds a system-defined minimum threshold T, the router begins dropping each new arriving packet with probability proportional to the load. After total load exceeds a system-defined maximum threshold, the packets are dropped with probability 1. (Note that when the maximum threshold is set to C, then once capacity is reached, RED behaves exactly like Droptail.)

To simplify our study, we will assume that the maximum threshold is C, but we will leave the minimum threshold T as a free parameter. We then must model the RED mechanism in the window game setup. Assume that the current load at the queue is $L \geq T$. Then, according to RED, each new arriving packet will be dropped with probability $\frac{L-T}{C-T}$. Assume that when W packets arrive sequentially, the expected number of them that enter the queue is x. In contrast to this sequential process where packets arrive one by one, using the window game we assume that given a strategy profile w, all W packets arrive at the same time. Each packet will be admitted to the queue with probability $\frac{x}{W}$ (x packets are chosen uniformly at random). If $W \leq T$, then all packets are admitted.

Lemma 4.1. *Assume that RED is used and let w be a strategy profile such that $W > T$.*

i) If $W \geq W_C$, where $W_C = (C-T)H_{C-T} + T$, then the queue size reaches C.

ii) If $T \leq W < W_C$, then $T + \tilde{k}_W$ packets enter the queue, (and the probability for any packet to be kept is $\frac{\tilde{k}_W + T}{W}$), where $\tilde{k}_W \approx (C-T)\left(1 - e^{-\frac{W-T}{C-T}}\right)$.

Proof. The proof uses the solution to the well-known coupon collector problem. In what follows we use the term *kept* to refer to the event of a packet not being dropped. We consider the case that W packets arrive sequentially. Consider the moment at which the queue size becomes $T+i-1$, for some $i, 1 \leq i \leq C-T$. Let X_i be a random variable that represents the number of packets that arrive to the system until the queue size reaches $T+i$ (i.e., $X_i - 1$ is the number of packets that arrive to the router and get dropped until one is kept). According to the description of RED, when $T+i-1$ packets are already in the queue, the probability that a newly arriving packet is dropped is $\frac{i-1}{C-T}$. This implies that $\mathbf{E}[X_i] = \frac{C-T}{C-T-i+1}$. Let H_j be the jth harmonic number.

i) $W_C = T + \mathbf{E}\left[\sum_{i=1}^{C-T} X_i\right] = T + \sum_{i=1}^{C-T} \frac{C-T}{C-T-i+1} = (C-T)H_{C-T} + T.$

ii) The total number number of packets $\tilde{k}_W$ that enter the queue, out of the total of W that arrive, is given as the maximum k, such that $T + \mathbf{E}\left[\sum_{i=1}^{k} X_i\right] \leq W \Leftrightarrow \sum_{i=1}^{k} \frac{C-T}{C-T-i+1} \leq W - T \Leftrightarrow (C-T)\left(H_{C-T} - H_{C-T-k}\right) \leq W - T$.

Approximating H_j with $\ln j$ we get: $\ln(C-T-k) \geq \ln(C-T) - \frac{W-T}{C-T}$ which gives $\tilde{k}_W \approx (C-T)\left(1 - e^{-\frac{W-T}{C-T}}\right)$. □

In order to simplify our presentation (and to allow clean formulation of a best response function), we will approximate $\tilde{k}_W$ by $k_W = \frac{W-T}{H_{C-T}}$. Note that k_W is also a continuous function, while $k_T = 0$ and $k_{W_C} = C - T$; therefore, when W equals T (respectively, W_C), the total number of packets entering the queue is T (respectively, C), in accordance to Lemma 4.1. The payoff function of flow i is now expressed as:

$$\pi_i^{RED}(w) = \begin{cases} w_i & \text{if } W \leq T \\ w_i \cdot \frac{k_W+T}{W} - g w_i\left(1 - \frac{k_W+T}{W}\right) & \text{if } T < W \leq W_C \\ w_i \cdot \frac{C}{W} - g w_i\left(1 - \frac{C}{W}\right) & \text{if } W > W_C \end{cases}$$

The best response function of RED differs according to the value of g. In particular, there are three possible ranges for g. Due to space limitations, we will only discuss here the case where $g \in R_g$, for $R_g = \left[\frac{C}{(C-T)(H_{C-T}-1)}, n-1\right]$, which is the most practically relevant range of values.[4] We defer the other cases, as well as the proofs of the following two theorems, to the full version of this work.

Definition 4.2. *Define* $r_g = \frac{T(g+1)(H_{C-T}-1)}{gH_{C-T}-g-1} \cdot \frac{n-1}{n^2}$.

Theorem 4.3. *If $g \in R_g$, then there is a unique NE, such that $w_i = r_g$, for all i.*

Theorem 4.4. *If $g \in R_g$, then the only stochastically stable state under RED is the state where all players set their window sizes to r_g.*

The above theorems imply that under RED the system will converge to the unique Nash Equilibrium. Given that $g \in R_g$, the total congestion will be less than the corresponding one in Droptail. Still, however, the overflow is large: as n grows, since $\frac{(g+1)(H_{C-T}-1)}{gH-g-1} > \frac{g+1}{g}$, the total window size will be (roughly) at least $2T$. And, as g decreases to values outside of R_g, the congestion at RED NE can sometimes be even greater than at the Droptail NE. More details can be found in the full version of this work.

5 "Fair" Queue Policy

In this section we study the queuing policy proposed by Gao et al. in [10]. The main idea (similar also to the Prince algorithm of [7]) is that in case of congestion, the most demanding flow is punished. Assuming that all players are fully informed of the other

[4] In practice $T = \lambda C$, for some constant λ meaning that $\frac{C}{(C-T)(H_{C-T}-1)}$ is a decreasing function on C tending to 0, as C grows large.

players' strategies, this policy was constructed so as to have a unique NE in which all players share the capacity equally. In a more realistic setting, where the rates at which other flows send packets are not globally known, the authors wish to reach a less lofty goal: if all flows but one have fixed rates, then the unrestricted flow cannot use up much more of the router queue capacity at the expense of the fixed-rate flows. We will show here that in fact, the fair equilibrium is also the only stochastically stable state. This implies that, even without fully informed players, the algorithm in [10] can achieve the fair NE, even when all flows are allowed to be arbitrarily aggressive.

The window game adaptation of Protocol I in [10] works as follows. For any profile $(w_1, \cdots, w_n)$, if $W \leq C$ then for any flow i, all w_i packets will enter the queue, i.e. $\pi_i(w) = w_i$. On the other hand, if $W > C$ then let $i_0 = \arg\max_{i \in N}\{w_i\}$ (breaking ties arbitrarily). Flow i_0 will be the one to be punished for the overflow, and if $w_{i_0} < W - C$ then the rest of the packets will be dropped according to Droptail. In other words, $\pi_{i_0} = \max\{0, w_{i_0} - (W - C)\} - g \cdot \min\{w_{i_0}, W - C\}$, while for any $i \neq i_0$,

$$\pi_i(w) = \begin{cases} w_i & \text{if } w_{i_0} \geq W - C \\ w_i \cdot \frac{C}{W - w_{i_0}} - g w_i \left(1 - \frac{C}{W - w_{i_0}}\right) & \text{if } w_{i_0} < W - C \, . \end{cases} \tag{2}$$

The next theorem was stated in [7].

Theorem 5.1. *Assuming $g > 0$, there is a unique NE in which all players play C/n.*

The following theorem establishes the fact that the unique NE is also the only stochastically stable state. We prove this by showing that the profile $(C/n, ..., C/n)$ is the only minimal cusber set.

Theorem 5.2. *If $g > 0$, then the only stochastically stable state is $(C/n, ..., C/n)$.*

Proof. Let $\hat{w} = (C/n, \cdots, C/n)$. We will show that the singleton set $\{\hat{w}\}$ is *the only* minimal cusber set. Then we can conclude using Corollary 2.3 that $\hat{w}$ is the only stochastically stable state. First note that $\{\hat{w}\}$ is a minimal cusber set: any player deviating from $\hat{w}$ will be strictly decreasing her payoff. (Assume that a player i moves to some value $x \neq C/n$. If $x < C/n$, then $\pi_i(x, \hat{w}_{-i}) = x < C/n = \pi_i(\hat{w})$. If $x > C/n$, then $x - C/n$ of her packets will be dropped and her payoff will decrease to $\pi_i(x, \hat{w}_{-i}) = C/n - g(x - C/n) < \pi_i(\hat{w})$, since $g > 0$.)

We proceed now to showing that for any profile $w \neq \hat{w}$, there is a finite better response path to $\hat{w}$. Assume first that $W > C$ and let $i_0 = \arg\max_{i \in N}\{w_i\}$. Then $\min\{w_{i_0}, W - C\}$ of i_0's packets get dropped. In that case it is at least as good for i_0 to play $\max\{0, w_{i_0} - (W - C)\}$, since the same amount of i_0's packets will enter the queue as before, but without any being dropped. We will call this a *move of type A*.

Assume now that $W = C$, but $w \neq (C/n, \cdots, C/n)$. Let j be the player with the maximum window size in w, i.e., $j = \arg_{i \in N} \max w_i$. The fact that $W = C$ and $w \neq \hat{w}$, imply that $w_j > C/n$. Moreover there must be some player $k \neq j$, with $w_k < C/n$. Playing C/n is a better response to k, since $C/n < w_j$, meaning that j will be the one to be punished for the overflow. (The new total window size cannot exceed the capacity by more than C/n, implying only packets from flow j will be dropped.) Therefore, k gets more packets in the queue by changing w_k to C/n, and still none dropped. We will call this a *move of type B*.

Now the better response path to $\hat{w}$ is constructed as follows: From any w, if $W < C$, then any player can improve her payoff by increasing her window size by $C - W$. We then arrive at a profile w' where $W' = C$. If $W > C$, after fewer than n moves of type A, a strategy profile w' is reached where $W' = C$.

For any w' such that $W' = C$, if $w' \neq \hat{w}$, then a move of type B occurs in which a player that in w' played something less than C/n moves to C/n. This is immediately followed by a move of type A in which a player that in w' was playing something greater than C/n reduces her window size. If w'' corresponds to the new profile reached, then again $W'' = C$. This alternation between moves of type A and moves of type B continues, until $\hat{w}$ is reached. Note that once a player moves to C/n then she does not change her window size anymore, meaning that the total number of steps needed until $\hat{w}$ is reached is finite. □

We note that the condition $g > 0$ in the above theorem is necessary in order for the cusber set to contain only the profile $(C/n, \ldots, C/n)$. If $g = 0$, then a flow can deviate from the profile $(C/n, \ldots, C/n)$ by increasing its window size while still obtaining exactly the same payoff. We also note that unlike the results of Sections 3 and 4, here, the result in this section holds even if each flow has a different value for g, a value that can be arbitrarily small.

6 Discussion

While Droptail and RED have stochastically stable states with high congestion at the bottleneck router, the Gao et al. policy leads to fair and efficient use of the bottleneck router capacity. Specifically, we've established that under Droptail queuing, the unique stochastically stable state (and unique NE) is the profile in which all players send a window size of $d_g = C(g+1)(n-1)/(gn^2)$. This means that if $g \leq (n-1)/(n+1)$, each player will be sending at least $2C/n$ packets, which amounts to twice as many total packets as the capacity allows.

Under RED, when g is reasonably large (i.e., for $g \in R_g$), the unique stochastically stable state (and unique NE) is the profile where all players send a window size of r_g, which is greater than $T(g+1)(n-1)/(gn^2)$. (Recall that $T < C$ is the threshold value at which RED begins preemptively dropping packets. It is a free parameter of the RED protocol.) This means, analogously to the above discussion about Droptail, that if $g \leq (n-1)/(n+1)$ (which is close to 1 as n grows large), players will be sending at least $2T/n$. This would imply that even with values of g nearly as large as 1, if deployers of RED routers set T to relatively large values, the gain with respect to overflow, as compared to the case of Droptail, will be small.

On the other hand, the more discriminating Gao et al. protocol can be safely deployed without knowledge of the specific value of g: the endpoints each send C/n as long as g is positive. In addition, our results hold even when each player has its own g value. Intuitively, this means the results apply even when the endpoints are all of different types: well-behaved TCP flows, more aggressive TCP flows, UDP flows, etc., as long as dropped packets cause some loss to the flows, no matter how small it is.

Finally we note the fact that the stochastically stable states in each case can be reached with the players having very limited knowledge; they need not be aware of the actions of other players, or even of their number n. Even though we assumed that players choose a window size between 0 and C, any other sufficiently large upper bound for the window sizes would have done just as well. In other words, the players need also not be aware of the exact value of the router capacity C.

Acknowledgments. The authors wish to thank Kirk Pruhs for invaluable advice and guidance, Lazaros Tsavlidis for his help with the window game, and Ihsan Qazi, Subrata Acharya, and Ho-Leung Chan for helpful discussions.

References

1. Akella, A., Seshan, S., Karp, R.M., Shenker, S., Papadimitriou, C.H.: Selfish behavior and stability of the internet: a game-theoretic analysis of tcp. In: SIGCOMM (2002)
2. Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C.L.P., Ramakrishnan, K.K., Shenker, S., Wroclawski, J., Zhang, L.: RFC2309: Recommendations on queue management and congestion avoidance in the internet. Internet RFCs (1998)
3. Chung, C., Ligett, K., Pruhs, K., Roth, A.: The price of stochastic anarchy. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 303–314. Springer, Heidelberg (2008)
4. Demers, A., Keshav, S., Shenker, S.: Analysis and simulation of a fair queueing algorithm. Applications, Technologies, Architectures, and Protocols for Computer Communication, 1–12 (1989)
5. Douligeris, C., Mazumdar, R.: On Pareto optimal flow control in an integrated environment. Allerton Conference on Communication, Control and Computing (1987)
6. Douligeris, C., Mazumdar, R.: A game theoretic approach to flow control in an integrated environment. Journal of the Franklin Institute 329(3), 383–402 (1992)
7. Efraimidis, P.S., Tsavlidis, L.: Window-games between TCP flows. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 95–108. Springer, Heidelberg (2008)
8. Floyd, S., Van Jacobson: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Netw. 1(4), 397–413 (1993)
9. Foster, D., Young, P.: Stochastic evolutionary game dynamics. Theoret. Population Biol. 38, 229–232 (1990)
10. Gao, X., Jain, K., Schulman, L.J.: Fair and efficient router congestion control. In: SODA 2004 (2004)
11. Garg, R., Kamra, A., Khurana, V.: A game-theoretic approach towards congestion control in communication networks. SIGCOMM Comput. Commun. Rev. 32(3), 47–61 (2002)
12. Jacobson, V.: Congestion avoidance and control. ACM SIGCOMM (1988)
13. Josephson, J., Matros, A.: Stochastic imitation in finite games. Games and Economic Behavior 49(2), 244–259 (2004)
14. Kesselman, A., Leonardi, S., Bonifaci, V.: Game-theoretic Analysis of Internet Switching with Selfish Users. In: Deng, X., Ye, Y. (eds.) WINE 2005. LNCS, vol. 3828, pp. 236–245. Springer, Heidelberg (2005)
15. Lee, H.J., Lim, J.T.: Performance Analysis of CHOKe with Multiple UDP Flows. In: SICE-ICASE, 2006. International Joint Conference, pp. 5200–5203 (2006)
16. Menasché, D.S., Figueiredo, D.R., de Souza e Silva, E.: An evolutionary game-theoretic approach to congestion control. Perform. Eval. 62(1-4), 295–312 (2005)

17. Pan, R., Prabhakar, B., Psounis, K.: Choke, a stateless active queue management scheme for approximating fair bandwidth allocation. In: INFOCOM (2000)
18. Shenker, S.J.: Making greed work in networks: a game-theoretic analysis of switchservice disciplines. IEEE/ACM Transactions on Networking 3(6), 819–831 (1995)
19. Stoica, I., Shenker, S., Zhang, H.: Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. In: ACM SIGCOMM 1998 (1998)
20. Tang, A., Wang, J., Low, S.H.: Understanding CHOKe: throughput and spatial characteristics. IEEE/ACM Transactions on Networking 12(4) (2004)
21. Peyton Young, H.: The evolution of conventions. Econometrica 61(1), 57–84 (1993)
22. Peyton Young, H.: Individual Strategy and Social Structure. Princeton University Press, Princeton (1998)

Nash Dynamics in Constant Player and Bounded Jump Congestion Games

Tanmoy Chakraborty and Sanjeev Khanna

Department of Computer and Information Science
University of Pennsylvania
tanmoy@cis.upenn.edu, sanjeev@cis.upenn.edu

Abstract. We study the convergence time of Nash dynamics in two classes of congestion games – constant player congestion games and bounded jump congestion games. It was shown by Ackermann and Skopalik [2] that even 3-player congestion games are PLS-complete. We design an FPTAS for congestion games with constant number of players. In particular, for any $\epsilon > 0$, we establish a stronger result, namely, any sequence of $(1+\epsilon)$-greedy improvement steps converges to a $(1+\epsilon)$-approximate equilibrium in a number of steps that is polynomial in ϵ^{-1} and the size of the input. As the number of strategies of a player can be exponential in the size of the input, our FPTAS result assumes that a $(1+\epsilon)$-greedy improvement step, if it exists, can be computed in polynomial time. This assumption holds in previously studied models of congestion games, including network congestion games [9] and restricted network congestion games [2].

For bounded jump games, where jumps in the delay functions of resources are bounded by β, we show that there exists a game with an exponentially long sequence of α-greedy best response steps that does not converge to an α-approximate equilibrium, for all $\alpha \leq \beta^{o(n/\log n)}$, where n is the number of players and the size of the game is $O(n)$. So in the worst case, Nash dynamics may fail to converge in polynomial time to such an approximate equilibrium. We also prove the same result for bounded jump network congestion games. In contrast, we observe that it is easy to show that a β^{2n}-approximate equilibrium is reached in at most n best response steps.

1 Introduction

A fundamental problem in algorithmic game theory is to determine the computational complexity of computing a *Nash equilibrium* for various classes of *non-cooperative games*. Nash [14] showed that a *mixed* Nash equilibrium always exists in any *finite* game, where there are finite number of players and strategies. In a mixed Nash equilibrium, players are allowed to play randomized strategies, and they wish to maximize their expected payoff. In contrast, pure Nash equilibrium, where players play deterministic strategies only, may not exist in all games. However, there are classes of games which always have a pure Nash equilibrium. The most prominent class with this property is *potential games*, defined by Monderer and Shapley [13].

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 196–207, 2009.

Nash equilibrium is especially interesting in systems where selfish agents make their own decisions, in a decentralized fashion. Nash dynamics is a well-studied model of selfish decentralized decision-making in a game, where players always play deterministic strategies (eg. [20]). The players either wish to maximize their payoff or minimize their cost – accordingly, the game is called profit maximizing or cost minimizing. The joint strategy of all the players is said to define a *state* of the game. Given a state of the game, a *better response* of a player is any of her strategies that has a higher payoff (or less cost) if the strategies of the other players remain unchanged. A *best response* refers to a strategy that has the highest payoff (or least cost, respectively) among the better responses. A pure Nash equilibrium is a state where no player has a better response. *Nash dynamics* is a repeated process in the game that starts from some *initial state*), and in each step, a player switches to a better response given the current state. We refer to each such step as an *improvement step*. Nash dynamics is also referred to as *best response dynamics* if the improvement steps are always best responses. Clearly, Nash dynamics stops (converges) when no player has a better response, that is, the dynamics has reached a pure Nash equilibrium.

Potential games [13] are defined as games where there is a potential function defined on the states of the game, which is finite-valued, and satisfies the property that for any improvement step, the change in potential is equal to the change in delay of the player that makes the step. A potential game may be profit maximizing or cost minimizing. Thus in a potential game, the Nash dynamics, starting from any state, always converges to a pure Nash equilibrium. However, it may take a long time to converge, possibly the number of states of the game. The time taken by Nash dynamics to converge in various potential games is the focus of this paper. We shall consider a worst-case analysis, that is, the longest possible sequence of improvement steps starting from any state. Potential games include various natural classes of games such as congestion games [16,9], cut games (or party affiliation games) [9,18,7] and market sharing games [10,21].

In particular, we shall focus on congestion games. A *congestion game* is an n-player game with m *resources*, and each player is assigned several strategies that she may choose from, where a strategy of a player is a subset of the resources. A player playing a strategy s is said to be *using* the resources in s. Each resource e is also associated with a non-negative increasing integral *delay function* $d_e(f_e)$, where f_e is the number of players using e. Each player that uses e suffers a delay of $d_e(f_e)$ on e, and the total delay of a player playing strategy s is $\sum_{e\in s} d_e(f_e)$. Every player must play one strategy, and selfishly seeks to *reduce* her own delay. Rosenthal [16] showed that every congestion game has a pure Nash equilibrium, and the proof effectively showed that every congestion game is a potential game. The potential function defined by Rosenthal is $\sum_e \sum_{i=1}^{c(e)} d_e(i)$ where $c(e)$ is the number of players using a resource e, known as the *congestion* of e.

Computational Complexity. Nash dynamics can be viewed as a natural algorithm to compute pure Nash equilibrium in potential games. However, the number of steps required to converge to an equilibrium from some initial state may not be polynomially bounded, and so this algorithm need not run in polynomial time. Johnson et al. [12] introduced a complexity class PLS related to local search problems, and Nash dynamics implies that computing a pure Nash equilibrium for potential games belong to this class, as long as the best-response can be computed in polynomial time (in the size of the game in a specific representation).

In fact, Fabrikant et al. [9] proved that the computing a pure Nash equilibrium in congestion games is PLS-complete. They also defined a subclass of congestion games called *network congestion games*, in which the input is a directed network, whose edges are the resources, and players correspond to source-sink pairs with strategies being the paths from the source to the sink. These games are also known as routing games with unsplittable flow, and are examples of games where the number of strategies may be exponential, but the best response can be computed in time polynomial in the size of the input. It was shown in [9] that computing equilibrium in network congestion games is also PLS-complete. Fabrikant et al. [9] also showed that computing equilibria in *symmetric* congestion games, where every player has the same strategy space, is PLS-complete. The PLS-completeness proofs for these problems also show that there exists initial states such that any sequence of improvement steps leading from the initial state to some equilibrium has exponentially many steps, regardless of the order of players chosen to make the improvement step.

Approximate Nash Equilibrium and Greedy Dynamics. Since it is considered to be unlikely that PLS = P, subsequent research has focused on computing approximate equilibria of congestion games. A natural definition for α-*approximate equilibrium* (see, eg. [17,7,1,11,6,19,3]) is a state where no player can unilaterally change her own strategy to reduce her own delay by a factor of α. In other words, a player is indifferent between strategies whose delays differ by less than a factor of α.

When looking for α-approximate equilibrium, it is natural to assume that all players shall make α-*greedy* improvement steps only, that is, a player may switch to a better response during Nash dynamics if the move reduces her delay by a factor of at least α. This α-greedy dynamics in potential games was first studied in [7].

1.1 Constant Player Congestion Games

The first result in this paper concerns potential games with constant number of players. In a potential game with constant number of players, if the strategies are explicitly specified in the input, then the number of states in the game is polynomial in the size of the input, so Nash dynamics reaches exact equilibrium in polynomial steps. Thus the question of determining whether the convergence time of Nash dynamics is polynomial, becomes interesting only when its representation is such that the number of strategies is exponential in the size of the input, but the best response can be computed in polynomial time. This condition is satisfied by network congestion games [9].

Ackermann and Skopalik [2] defined a class of games called *restricted network congestion games* , which is a network congestion game where each player is barred from using a certain subset of edges. These games are potential games. Moreover, they are congestion games, and the best response of a player can be computed in polynomial time. Ackermann and Skopalik [2] showed that computing an equilibrium in restricted network congestion games with only 3 players is PLS-complete.

This result can be compared with the problem of computing mixed Nash equilibrium in normal-form games, which is PPAD-complete, and moreover, an FPTAS cannot exist even for two-player games unless PPAD = P, which is unlikely [15,8,4,5]. In contrast, we give an FPTAS for computing a pure Nash equilibrium in all congestion games with constant number of players where the best response can be computed in polynomial time, and the input size is at least logarithm of the maximum potential of any state. The latter assumption is justified if the delay functions are written explicitly in the input. This includes network congestion games [9] and restricted network congestion games [2]. We show this by proving the stronger result that all sequences of improvement steps are polynomial in length. The formal statement of our result is given by the theorem below.

Theorem 1. *In any congestion game with k players and an arbitrary initial state s with potential $\varphi(s)$, every sequence of $(1+\epsilon)$-greedy steps reaches a $(1+\epsilon)$-approximate equilibrium in at most $(2k\epsilon^{-1}\ln\varphi(s))^{2k}$ steps, for any $0 < \epsilon \leq 1$.*

Note that an FPTAS is immediately implied by the above theorem if it can be determined in polynomial time whether a given state is a $(1+\epsilon)$-approximate equilibrium, and if it is not, a $(1+\epsilon)$-greedy step can also be computed in polynomial time. Our proof easily implies that the above result is also true for profit-maximizing potential games with payoff for each resource, such as market sharing games [10,21].

1.2 Bounded Jump Congestion Games

Chien and Sinclair [6] gave an FPTAS for symmetric congestion games with polynomially *bounded jumps*. They showed that several natural variants of Nash dynamics converges fast in such games. However, they could not show that all sequences converge fast to an approximate equilibrium, unlike Theorem 1. However, their result handles arbitrary number of players, although requiring the bounded jump condition and symmetry, unlike our result. A resource e is said to satisfy the *bounded jump condition* if its delay increases by a factor of at most β with the addition of a new player using e, for some polynomially bounded parameter β.

Progress has also been made on the inapproximability of equilibria in congestion games (via Nash dynamics or otherwise). Ackermann et al. [1] constructed an asymmetric congestion game with bounded jumps, in which there exists an initial state and an exponentially long sequence of α-greedy steps leading to an α-approximate equilibrium. This result was strengthened in [19], with the construction of an asymmetric congestion game with bounded jumps, where there exists an initial state such that every sequence of α-greedy steps from this state to some α-approximate equilibrium is exponential in length. Finally, and most recently, [19] also showed that for asymmetric congestion games with unbounded jumps (that is, general congestion games), computing α-approximate equilibrium is PLS-complete for any polynomially computable α. We note that this result is true for symmetric congestion games with unbounded jumps as well.

The construction in [19] for (asymmetric) congestion games with β-bounded jumps does not allow polynomially bounded sequences of α-greedy steps leading to an α-approximate equilibrium, where $\beta = \Theta(\alpha^{27})$. Thus, it leaves open the question whether the Nash dynamics converges in polynomial steps to a γ-approximate equilibrium in β-bounded congestion games, for some γ that is polynomial in β. Given that general congestion games are hard to approximate within any polynomial factor [19], and that the reductions critically use resources with unbounded jumps, bounded jumps is one of the most natural restrictions one can put on the game and hope for positive results about computing approximate equilibrium. However, we provide negative evidence for it, showing that Nash dynamics may fail badly in the worst case. We also construct a network congestion game that exactly model the constructed congestion game mentioned above, and thus obtain the same result for network congestion games with uniform jump.

Theorem 2. *For every $\beta > 1$, there exists infinitely many integers n, such that there exists a congestion game $G(n)$ with n players and $2n$ resources, and with jumps bounded by β that satisfies the following property. For every $\alpha \leq \beta^{o(n/\log n)}$, there exists an initial state of the game and a super-polynomial length sequence of α-greedy best response steps, such that the resulting state is not an α-approximate equilibrium.*

In fact, there exists a network congestion game with $O(n)$ players and $O(n)$ vertices and edges, and with jumps bounded by β, satisfying the same property.

It is easy to see that in any congestion game where the jumps are bounded by β, a sequence of β^{2n}-greedy best response steps starting from any state, leading to a β^{2n}-approximate equilibrium, is of length at most n – each player can move at most once. Our negative result indicates that without a smart choice of the order in which players make improvement steps, in the worst case, Nash dynamics shall not converge in polynomial time for any approximation that is significantly better than the trivially obtainable β^{2n}. In our construction, every player has only two strategies, so there is no distinction between best response and better response Nash dynamics. Further, all resources in our construction satisfy a stronger condition that we call the *uniform jump condition*, where the delay increases by a factor of *exactly* β with the addition of a new player using the resource, that is, $d_e(t+1) = \beta d_e(t)\ \forall t \geq 1$. This is interesting because such delay functions can be represented succinctly by just two integers, β and $d_e(1)$.

Organization. The rest of the paper is organized as follows: we formally define congestion games and its various restrictions in Section 2. In Section 3, we prove our result for constant player congestion games, namely, Theorem 1. In Section 4, we describe our construction of a uniform jump general congestion game that proves Theorem 2.

2 Preliminaries

A *game* consists of a finite set of players $p_1, p_2 \dots p_n$. Each player p_i has a finite set of strategies S_i and a *cost or delay* function $c_i : S_1 \times \dots S_i \times \dots S_n \to \mathbb{N}$ that it wishes to minimize. A game is called *symmetric* if all strategy sets S_i's are identical. A combination of strategies $s = (s_1, s_2 \dots s_n) \in S_1 \times \dots S_i \times \dots S_n$ is called a *state* of the game, where p_i plays strategy $s_i \in S_i$. A state s is a *pure Nash equilibrium* if for all players p_i, $c_i(s_1, ..., s_i, ..., s_n) \leq c_i(s_1, ..., s'_i, ..., s_n)$ for all $s'_i \in S_i$. In such a state, no player can improve its cost by unilaterally changing its strategy.

Congestion games is a class of games where players' costs are based on the shared usage of a common set of resources $E = \{e_1, e_2 \dots e_m\}$. Each strategy of a player is a subset of resources, and the strategy set of p_i is $S_i \subseteq 2^E$, an arbitrary collection of subsets of E. Each resource $e \in E$ has a non-decreasing delay function $d_e : 1, ..., n \to \mathbb{N}$ associated with it. If j players are using the resource e, each of these players incurs a delay of $d_e(j)$. The *delay* incurred by a player p_i in a state $s = (s_1, \dots s_n)$ is the sum of the delays it incurs on each resource in its strategy, that is, $c_i(s) = \sum_{e \in s_i} d_e(f_s(e))$, where $f_s(e)$ is the number of players using resource e in state s, that is, $f_s(e) = |\{j : e \in s_j\}|$. A *network congestion game* is a congestion game defined over an underlying network such that edges of the network are the resources and the strategies correspond to paths in the network. Specifically, each player p_i is associated with a source vertex s_i and a sink vertex t_i. The strategy set of a player p_i is the set of all paths from s_i to t_i.

The following function, defined on a state $s = (s_1 \dots s_i \dots s_n)$ of a congestion game is called the *potential function* of the game: $\varphi(s) = \sum_e \sum_i^{f_s(e)} d_e(i)$. This function has the property that if p_i changes its strategy from s_i to s'_i, all other players' strategies remaining unchanged, then the potential of the resulting state $s' = (s_1 \dots s'_i \dots s_n)$ is $\varphi(s') = \varphi(s) + (c_i(s') - c_i(s))$, that is, the change in potential is equal to the change in the delay of p_i [16].

Given a state $s = (s_1 \dots s_n)$, a *better response* strategy of a player p_i is some strategy $s'_i \in S_i$ such that if the player switches her strategy from s_i to s'_i her delay decreases. A *best response* is a better response strategy that maximizes this decrease. Changing the state by making a player switch to a better response strategy is called an *improvement step*. *Nash dynamics* starting from some state s refers to a sequence of states such that only one player changes strategy in one step, and each such change is an improvement step with respect to the preceding state.

An α*-approximate equilibrium* is a state s such that for any strategy $s'_i \in S_i$, if $s' = (s_1, s_2 \dots s'_i \dots s_n)$, then $c_i(s) \leq \alpha c_i(s')$, for all $1 \leq i \leq n$. An α*-greedy* improvement step is an improvement step where the change of strategy causes the player's delay to decrease by a multiplicative factor of at least α, that is, if state s changed to state s', $c_i(s') \leq c_i(s)/\alpha$.

A resource e is said to satisfy the *bounded jump* condition if $d_e(i+1)/d_e(i) \leq \beta \; \forall i \geq 1$ for some β. In this case, the jumps are said to be bounded by β. A resource e is said to satisfy the *uniform jump* condition if $d_e(i+1) = \beta d_e(i) \; \forall i \geq 1$ for some β. Note that for every resource e satisfying the uniform jump condition, and every positive integer i, we have $d_e(i) = \beta^{(i-1)} d_e(1)$. So the values $d_e(1)$ and β succinctly define the entire delay function.

3 Constant Player Congestion Games

In this section, we prove Theorem 1. Let $a(k, s)$ be the maximum number of $(1+\epsilon)$-greedy steps required by any congestion game of k players to converge to a $(1+\epsilon)$-approximate equilibrium starting from state s. We shall prove by induction on the number of players k that $a(k, s) \leq (2k\epsilon^{-1} \ln \varphi(s))^{2k}$.

As the base step for induction, let $k = 1$. The potential function is simply the delay of the lone player. Since in each step, the lone player must improve her delay by a factor of $(1+\epsilon)$, the potential after t steps is at most $\varphi(s)/(1+\epsilon)^t$. Since the delay functions are all integral, once the delay is less than 1, it must be zero and the equilibrium must be reached. Thus, an upper bound on the number of steps is the maximum value of t such that $\varphi(s)/(1+\epsilon)^t \geq 1$, which implies that $t \leq \log_{1+\epsilon} \varphi(s) = \frac{\ln \varphi(s)}{\ln(1+\epsilon)}$. Since $\epsilon \leq 1$, so $\ln(1+\epsilon) > \epsilon/2$, and we have $a(1, s) \leq 2\epsilon^{-1} \ln \varphi(s)$, thus proving the base case.

For the inductive step, assume that the assertion holds for all congestion games with $(k-1)$ players. Now, consider any congestion game with k players $p_1, p_2 \ldots p_k$. Let us consider any state $s = (s_1, s_2 \ldots s_k)$ where s_i is the strategy of player p_i. Without loss of generality, let p_k be the player with the highest delay $C = \sum_{e \in s_k} d_e(c_s(e))$, where $c_s(e)$ is the congestion on resource e in state s. Note that the sum of the delays of all players is $\sum_e c_s(e) d_e(c_s(e)) \geq \sum_e \sum_{i=1}^{c_s(e)} d_e(i) = \varphi(s)$, since the delay functions are increasing. So $C \geq \varphi(s)/k$.

Now suppose we start our sequence of $(1+\epsilon)$-greedy steps from s. We need to put an upper bound on the number of steps that can happen before p_k must get to move or the process terminates. Observe that if the strategy of p_k remains fixed, the rest of the players are effectively playing a reduced game involving $k-1$ players, and the delay functions of the resources in s_k have been modified (by setting $d'_e(i) = d_e(i+1)$). It is easy to see that the potential of this state s^{red} of the reduced $(k-1)$-player game is no more than $\varphi(s)$. Thus the number of $(1+\epsilon)$-greedy steps that can occur before p_k moves or process terminates, is at most $a(k-1, s^{red})$, which by our induction hypothesis is at most $(2(k-1)\epsilon^{-1} \ln \varphi(s^{red}))^{2(k-1)} < (2k\epsilon^{-1} \ln \varphi(s))^{2(k-1)}$.

Now, when p_k finally moves, let s' denote the state that results after p_k makes a $(1+\epsilon)$-greedy step. The following claim is similar to that of Lemma 4.2 of [6], and we omit its proof due to lack of space. It should be noted that the definition of ϵ-approximation in [6] is in fact a $\frac{1}{1-\epsilon}$-approximation in our notation, which is the same as that in [19]. The difference in the definitions vanish when we seek an FPTAS.

Claim. $\varphi(s) - \varphi(s') \geq \frac{\epsilon}{1+\epsilon} C \geq \frac{\epsilon \varphi(s)}{k(1+\epsilon)}$. Since $\epsilon \leq 1$, so $\varphi(s') \leq \frac{\varphi(s)}{1+(\epsilon/2k)}$.

From the claim and the upper bound established previously, it follows that in $(2k\epsilon^{-1} \ln \varphi(s))^{2(k-1)} + 1$ steps, the potential must go down by a factor of $\left(1 + \frac{\epsilon}{2k}\right)$. Again, since the potential must be greater than 1 for an improvement step to exist, we have that

$$
\begin{aligned}
a(k,s) &\le \frac{\ln\varphi(s)}{\ln(1+\frac{\epsilon}{2k})}((2k\epsilon^{-1}\ln\varphi(s))^{2(k-1)}+1) \\
&\le \frac{\ln\varphi(s)}{\epsilon/4k}((2k\epsilon^{-1}\ln\varphi(s))^{2(k-1)}+1) \qquad \text{(since } \epsilon/2k \le 1) \\
&< (2k\epsilon^{-1}\ln\varphi(s))^{2k}
\end{aligned}
$$

4 Exponential Dynamics in Uniform (Bounded) Jump Congestion Games

In this section, we present the proof of first part of Theorem 2 by constructing a bounded jump congestion game that satisfies the conditions of the theorem. To complete the proof of Theorem 2, we show that the game constructed here can be modeled as a network congestion game. This latter part of the proof is deferred to the full version of the paper due to lack of space.

Let $\delta > 1$. We will construct a congestion game where every resource has a uniform jump of δ^4, that is, every resource e and positive integer i satisfies $d_e(i) = \delta^{4(i-1)}d_e(1)$. We shall say that a resource with such a delay function satisfies the *uniform jump condition* with jump factor δ^4. Let k be any integer such that $\delta^{k/2} > k \ge 2$, and let $r \ge 10$.

Let n_0 be any positive integer. Our constructed game has $4kn_0$ players, and $8kn_0$ resources. Also, every player has exactly two strategies. We shall exhibit a state s and a sequence of $\Theta(k2^{n_0})$ improvement steps, starting from s, such that each decreases the delay of the player by a factor of at least $\delta^{k/2}$. Thus $\delta^{k/2}$-greedy Nash dynamics will fail to produce a $\delta^{k/2}$-equilibrium in polynomial (in n_0) number of steps if the sequence of players is adversarially chosen, with each player choosing its best response when given an opportunity.

Assuming that we can construct such a game, we shall now prove Theorem 2 with a careful choice of the parameters k, n_0 and δ. Suppose we are given $\beta > 1$. We choose δ such that $\beta = \delta^4$. Then, every resource has a delay function with uniform jump β. Suppose we are also given some function $g(t) = o(t/\log t)$. We define a function $h(t) = \Theta(t/g(t)) = \omega(\log t)$ such that $h(t)$ is always a positive integer. For a particular t, we choose $n_0 = h(t)$, and we choose $k = \lceil t/n_0 \rceil = \Theta(g(t))$. Then the size of the game is $\Theta(kn_0) = \Theta(t)$, and $\delta^{k/2} = \beta^{\Theta(g(t))}$. Moreover, the length of the sequence of improvement steps is $\Theta(k2^{n_0}) = 2^{\omega(\log t)}$, which is superpolynomial in the length of the input, which is $O(t)$. This yields Theorem 2. Thus, it only remains to construct a game with the parameters δ, k and n_0 as mentioned above.

Description of the Game. The Nash dynamics will simulate an n_0-bit counter. Each bit of the counter is represented by $4k$ players which logically belong to 4 different groups of k players each. Also, there are $8k$ resources logically attributed to each bit. For the i^{th} bit of the counter, these resources are $a_{ij}, a'_{ij}, b_{ij}, b'_{ij}, u_{ij}, u'_{ij}, t_{ij}, t'_{ij}$, where $j = 1, 2 \ldots k$. The groups of players are as follows:

- $Tr_{i1}, Tr_{i2} \ldots Tr_{ik}$. These are the *first set of trigger players*. Tr_{ij} have one private resource t_{ij} that constitutes one of their strategies, $1 \leq j \leq k$. The other strategy is $\{a_{i1}, a_{i2} \ldots a_{ik}, u_{ij}\}$. We refer to playing the former strategy as the *reset state*, while the latter as the *triggered state*.
- $Tr'_{i1}, Tr'_{i2} \ldots Tr'_{ik}$. These are the *second set of trigger players*. Tr'_{ij} have one private resource t'_{ij} that constitutes one of their strategies, $1 \leq j \leq k$. The other strategy is $\{a'_{i1}, a'_{i2} \ldots a'_{ik}, u'_{ij}\}$. We refer to playing the former strategy as *reset state*, while the latter as *triggered state*.
- $p_{i1}, p_{i2} \ldots p_{ik}$. These are the *first set of base players*. One strategy for p_{ij} is $\{a_{ij}\}$, and this strategy is congested whenever the first set of trigger players get into triggered state. The other strategy is $\{b_{ij}\} \cup \{u'_{lq} \mid 1 \leq l < i, 1 \leq q \leq j\}$, in which it congests the triggered states of the second set of trigger players of all bits lower than i.
- $p'_{i1}, p'_{i2} \ldots p'_{ik}$. These are the *second set of base players*. One strategy for p'_{ij} is $\{a'_{ij}\}$, and this strategy is congested whenever the second set of trigger players get into triggered state. The other strategy is $\{b'_{ij}\} \cup \{u_{iq} \mid 1 \leq q \leq j\}$, in which it congests the triggered states of the first set of trigger players of the i^{th} bit.

Table 1. Strategies of players

Player(p)	First strategy $s(p,1)$ (private)	Second strategy $s(p,2)$ (non-private)
Tr_{ij}	t_{ij}	$a_{i1}, a_{i2} \ldots a_{ik}, u_{ij}$
Tr'_{ij}	t'_{ij}	$a'_{i1}, a'_{i2} \ldots a'_{ik}, u'_{ij}$
p_{ij}	a_{ij}	$u'_{11} \ldots u'_{1k}, u'_{21} \ldots u'_{2k}, \ldots u'_{(i-1)1} \ldots u'_{(i-1)k}, b_{ij}$
p'_{ij}	a'_{ij}	$b'_{ij}, u_{i1}, u_{i2} \ldots u_{ik}$

Interpreting Strategies as a Counter Value. We say that a player playing a singleton strategy is in its *private state*, and else to be in their *non-private state*. In Table 1, we describe the two strategies of every player. i ranges from 1 through n_0, while j ranges from 1 through k. We shall say that the i^{th} *bit of the counter is zero* if all players corresponding to the i^{th} bit are in their private states. We shall say that the i^{th} *bit of the counter is one* if all players in the first sets of trigger and base players are in their private state, while all players in the second sets of trigger and base players are in their non-private state. We shall also use the notation $e(i)$ to denote the delay on a resource e when i players are using it.

Overview. Our starting state corresponds to all bits in the counter initialized to zero. We implement a sequence of phases where each phase consists of a sequence of length $\Theta(k)$ such that (i) each step corresponds to an α-greedy improvement step for some player, and (ii) the resulting state corresponds to incrementing the counter by 1. The game terminates when all bits in the counter are set to 1, thus giving us an overall sequence of length $\Theta(k2^{n_0})$.

Implementation of a Phase. We now describe the implementation of a phase. Suppose at the beginning of the phase, the state corresponds to the bit i of the counter set to 0, and all lower-order bits set to 1. The sequence of improvement steps for this phase will convert the i^{th} bit to one and all lower bits to zero. The steps are as given below, along with the change in delay for the improving player (we shall specify the delay functions later):

1. The first set of triggers (in any order) of the i^{th} bit switches to their non-private state (triggered) and thus congests the private states of the first set of base players. Thus the state changes for Tr_{ij} as follows: $t_{ij}(1) \rightarrow \sum_{q=1}^{k} a_{iq}(w+1)+u_{ij}(1)$ for some $1 \leq w \leq k$. Note that w is solely determined by the order in which these trigger players are chosen to move. For example, if the natural order of $t_{i1}, t_{i2} \ldots t_{ik}$ is used, then $w = j$. A similar statement about the value of w holds for the steps below.
2. The first set of base players (in any order) move to their non-private states, thus congesting the second set of triggers of lower bits (who are in their non-private states). Thus the state changes for p_{ij} as follows: $a_{ij}(k+1) \rightarrow b_{ij}(1) + \sum_{l=1}^{i-1} \sum_{q=1}^{k} u'_{lq}(w+1)$ for some $1 \leq w \leq k$.
3. The second set of triggers of lower bits (in any order) move to their private states (they are reset), thus releasing the private resources of the corresponding base players. Thus the state changes for Tr'_{lj} as follows: $\sum_{q=1}^{k} a'_{lq}(w) + u'_{lj}(k+1) \rightarrow t'_{lj}(1)$ for some $1 \leq w \leq k$.
4. The second set of base players of lower bits (in any order) move to their private states too. Thus the lower bits have been converted to zeroes. Thus the state changes for p'_{lj} as follows: $b'_{lj}(1) + \sum_{q=1}^{k} u_{lq}(w) \rightarrow a'_{lj}(1)$ for some $1 \leq w \leq k$.
5. The second set of triggers of the i^{th} bit (in any order) now moves to their non-private state, congesting the private states of the second set of base players. Change for Tr'_{ij}: $t'_{ij}(1) \rightarrow \sum_{q=1}^{k} a'_{iq}(w+1) + u'_{ij}(1)$ for some $1 \leq w \leq k$.
6. The second set of base players (in any order) move to their non-private state, congesting the non-private state of the first set of triggers. Thus the state changes for p'_{ij} as follows: $a'_{ij}(k+1) \rightarrow b'_{ij}(1) + \sum_{l=1}^{i-1} \sum_{q=1}^{k} u_{iq}(w+1)$ for some $1 \leq w \leq k$.
7. The first set of triggers move to their private state (reset), releasing the private states of the first set of base players. Thus the state changes for Tr_{ij} as follows: $\sum_{q=1}^{k} a_{iq}(w) + u_{ij}(k+1) \rightarrow t_{ij}(1)$ for some $1 \leq w \leq k$.
8. The first set of base players move to their private states too, and the i^{th} bit has been converted to one. Thus the state changes for p_{ij} as follows: $b_{ij}(1) + \sum_{l=1}^{i-1} \sum_{q=1}^{k} u'_{lq}(w) \rightarrow a_{ij}(1)$ for some $1 \leq w \leq k$.

The states held by the players logically assigned to the i^{th} bit or less significant bits change through these steps, as expressed in Table 2, while players assigned to higher bits do not change their strategy. P stands for private strategy, N stands for non-private strategy of a player. Column labelled x shows the strategy of various players after the group of steps x in the sequence above. j varies from 1 to k, and l varies from 1 to $i-1$.

We shall now define the delay functions. Let $\gamma = \delta^k$. Since every resource has a uniform jump of δ^4, it suffices to simply define $e(1)$ for every resource e. Also, note that $r = 10$. The delay functions are defined in Table 3. Note

Table 2. Sequence of state vectors

Player	Initial	1	2	3	4	5	6	7	8 (Final)
Tr_{ij}	P	N	N	N	N	N	N	P	P
p_{ij}	P	P	N	N	N	N	N	N	P
Tr'_{ij}	P	P	P	P	P	N	N	N	N
p'_{ij}	P	P	P	P	P	P	N	N	N
Tr_{lj}	P	P	P	P	P	P	P	P	P
p_{lj}	P	P	P	P	P	P	P	P	P
Tr'_{lj}	N	N	N	P	P	P	P	P	P
p'_{lj}	N	N	N	N	P	P	P	P	P

Table 3. Delays of strategies

Resource(e)	a_{ij}	a'_{ij}	b_{ij}	b'_{ij}	u_{ij}	u'_{ij}	t_{ij}	t'_{ij}
$e(1)$	γ^{2ri}	$\gamma^{r(2i+1)}$	γ^{2ri+2}	$\gamma^{r(2i+1)+2}$	γ^{2ri+4}	$\gamma^{r(2i+1)+4}$	γ^{2ri+6}	$\gamma^{r(2i+1)+6}$

that $a_{ij}(k+1) = \gamma^{2ri+4}$, $a'_{ij}(k+1) = \gamma^{r(2i+1)+4}$, $u_{ij}(k+1) = \gamma^{2ri+8}$, and $u'_{ij}(k+1) = \gamma^{r(2i+1)+8}$.

Now we can verify that every improvement step is an improvement by a factor of more than γ/k, using the facts that $\gamma \geq 2$ and $r = 10$. By our choice of k, we have $\gamma/k > \gamma^{1/2} = \delta^{k/2}$.

Finally, we note that if all the bits are in their one state, then we have an exact equilibrium. To see this, note that in this state, players logically assigned to the i^{th} bit use resources logically assigned to this bit only. Thus, it is enough to check that if the game is restricted to just one bit, then the four groups of players assigned to this bit are at equilibrium. This is easy to check: The second set of trigger players have the higher delay on their private strategy, so it is best for them to play their non-private strategy, and so the second set of base players will be dissuaded from switching to their private strategies. Since the second set of base players are using their non-private strategies, the best-response of the first set of trigger players is to use their private strategies, and so the first set of base players can stay at their uncongested private strategy, which they prefer. Thus the Nash dynamics terminates when this state, where all bits are one, is reached. This completes the construction and analysis of the game.

We can also construct a network congestion game that captures the behavior of the congestion game in Section 4, thus completing the proof of Theorem 2. The construction of this network is omitted due to lack of space.

References

1. Ackermann, H., Röglin, H., Vöcking, B.: On the impact of combinatorial structure on congestion games. In: FOCS, pp. 613–622 (2006)
2. Ackermann, H., Skopalik, A.: On the complexity of pure nash equilibria in player-specific network congestion games. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 419–430. Springer, Heidelberg (2007)

3. Awerbuch, B., Azar, Y., Epstein, A., Mirrokni, V.S., Skopalik, A.: Fast convergence to nearly optimal solutions in potential games. In: ACM Conference on Electronic Commerce, pp. 264–273 (2008)
4. Chen, X., Deng, X.: Settling the complexity of two-player nash equilibrium. In: FOCS, pp. 261–272 (2006)
5. Chen, X., Deng, X., Teng, S.-H.: Computing nash equilibria: Approximation and smoothed complexity. In: FOCS, pp. 603–612 (2006)
6. Chien, S., Sinclair, A.: Convergence to approximate nash equilibria in congestion games. In: SODA, pp. 169–178 (2007)
7. Christodoulou, G., Mirrokni, V.S., Sidiropoulos, A.: Convergence and approximation in potential games. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 349–360. Springer, Heidelberg (2006)
8. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a nash equilibrium. In: STOC, pp. 71–78 (2006)
9. Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure nash equilibria. In: STOC, pp. 604–612 (2004)
10. Goemans, M.X., Li, E.L., Mirrokni, V.S., Thottan, M.: Market sharing games applied to content distribution in ad-hoc networks. In: MobiHoc, pp. 55–66 (2004)
11. Goemans, M.X., Li, L., Mirrokni, V.S., Thottan, M.: Market sharing games applied to content distribution in ad hoc networks. IEEE Journal on Selected Areas in Communications 24(5), 1020–1033 (2006)
12. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? J. Comput. Syst. Sci. 37(1), 79–100 (1988)
13. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14, 124–143 (1996)
14. Nash, J.F.: Noncooperative games. Annals of Mathematics 54, 289–295 (1951)
15. Papadimitriou, C.H.: On the complexity of the parity argument and other inefficient proofs of existence. J. Comput. Syst. Sci (also FOCS 1990) 48(3), 498–532 (1994)
16. Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. International Journal of Game Theory 2, 65–67 (1973)
17. Roughgarden, T., Tardos, É.: How bad is selfish routing? J. ACM 49(2), 236–259 (2002)
18. Schäffer, A.A., Yannakakis, M.: Simple local search problems that are hard to solve. SIAM J. Comput. 20(1), 56–87 (1991)
19. Skopalik, A., Vöcking, B.: Inapproximability of pure nash equilibria. In: STOC, pp. 355–364 (2008)
20. Tardos, E., Wexler, T.: Network formation games and the potential function method. In: Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. (eds.) Algorithmic Game Theory, ch. 19. Cambridge University, Cambridge (2007)
21. Vetta, A.: Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In: FOCS, p. 416 (2002)

Price of Stability in Survivable Network Design

Elliot Anshelevich and Bugra Caskurlu

Computer Science Department, RPI, 110 8th Street, Troy, NY 12180
{eanshel,caskub}@cs.rpi.edu

Abstract. We study the survivable version of the game theoretic network formation model known as the Connection Game, originally introduced in [4]. In this model, players attempt to connect to a common source node in a network by purchasing edges, and sharing their costs with other players. We introduce the *survivable* version of this game, where each player desires 2 edge-disjoint connections between her pair of nodes instead of just a single connecting path, and analyze the quality of exact and approximate Nash equilibria. For the special case where each node represents a player, we show that Nash equilibria are guaranteed to exist and price of stability is 1. For the general version of the Survivable Connection Game, we show that there always exists a 2-approximate Nash equilibrium that is as cheap as the socially optimal solution.

1 Introduction

The global performance of networks such as the Internet, which are developed, built, and maintained by a large number of selfish agents, may not be as good as in the case where a central authority can simply dictate a solution. In order to understand the performance of such systems, we need to analyze the quality of solutions that are consistent with self-interested behavior. Much research in the theoretical computer science community has focused on this performance gap and specifically on the notions of the *price of anarchy* and the *price of stability* — the ratios between the costs of the worst and best Nash equilibrium[1], respectively, and that of the centralized optimal solution. Both of these notions are important since they quantify the borders of the quality spectrum of Nash equilibria, which are often the only viable outcomes of agent interactions. We will only consider pure (i.e., deterministic) Nash equilibria, as mixed strategies do not make as much sense in our network design context.

Connection Game. In this paper, we consider the price of stability of several important extensions of the *Connection Game*, which was first defined in [4], and later studied in a variety of papers including [2,8,10,12,14,15]. This game represents a general framework where a network is being built by many different agents/players who have different connectivity requirements, but can combine

[1] Recall that a (pure-strategy) Nash equilibrium is a solution where no single player can switch her strategy and become better off, given that the other players keep their strategies fixed.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 208–219, 2009.

their money to pay for some part of the network. The Connection Game models not only communication networks, but also many kinds of transportation networks that are built and maintained by competing interests. Specifically, each player in this game has some connectivity requirements in a graph $G = (V, E)$, i.e, she desires to connect a particular pair of nodes in this graph. With this as their goal, players can offer payments indicating how much they will contribute towards the purchase of each edge in G. If the players' payments for a particular edge e sum to at least the cost of e, then the edge is considered *bought*, which means that e is added to our network and can now be used to satisfy the connectivity requirements of any player.

Survivable Network Design. One of the most important extensions of the Steiner Forest problem is Survivable Network Design (sometimes called the Generalized Steiner Forest problem). In this problem, we must not simply connect all the desired pairs of terminals, but instead connect them using r edge-disjoint paths. This is generally needed so that in the case of a few edge failures, all the desired terminals still remain connected. Many nice results have been shown for finding the cheapest survivable network, including Jain's 2-approximation algorithm [17].

In this paper, we consider the *Survivable Connection Game,* where each agent/ player wishes to connect to her destination using $r = 2$ edge-disjoint paths. The optimal (i.e., cheapest) centralized solution for this game is the optimal solution to Survivable Network Design, which we denote by OPT. Our major goal is to understand the level of consistency of selfish acts with global system efficiency, by comparing the cost of exact and approximate Nash equilibria to OPT.

Our Results. We only consider the case where all players are attempting to connect to a single common source. For the single source version of the Connection Game, [4] proved that the price of stability is 1, and that in particular, a pure Nash equilibrium always exists. This is no longer true if the players have arbitrary connection requirements, as a pure Nash equilibrium is no longer guaranteed to exist. As we show in the full version of the paper, a Nash equilibrium is not guaranteed to exist even for 2-player games, and there may not exist an α-approximate[2] Nash equilibrium for any $\alpha < 2$. So, adding stronger connectivity requirements to the Connection Game significantly changes it, since the prices of anarchy and stability become infinite. Instead of considering arbitrary connection requirements, therefore, we restrict our attention to the case where all terminals desire to connect using 2 disjoint paths. In this case, we prove results that are similar to the properties of the original Connection Game. Specifically, our main results are as follows:

– In the special case where all nodes are terminal nodes (i.e., there exists a player that desires to connect this node to the source), there always exists a Nash equilibrium that is as good as OPT.

[2] An α-approximate Nash equilibrium is a solution where no player can save more than a factor of α by deviating.

– For the general Survivable Connection Game, there exists a 2-approximate Nash equilibrium that is as good as OPT and there is a polynomial time algorithm which finds a cheap $(2+\epsilon)$-approximate Nash equilibrium.

Our approach for forming equilibrium payments is similar to [4], in that we show that either every edge can be bought using our equilibrium payment scheme, or that OPT is not actually an optimal solution, thus deriving a contradiction. The fact that OPT is no longer a tree, however, significantly complicates matters, forcing the payment scheme to be a bit more clever and requiring different proof techniques. In order to prove the results:

– We consider a version of the Survivable Connection Game where each player is only allowed to deviate by changing the payments on a single path and show that for that version, there always exists a stable solution that is as good as the centralized optimal solution, i.e, the price of stability is 1. We obtain our main result by proving that any stable solution of this version corresponds to a 2-approximate Nash equilibrium of the general Survivable Connection Game.
– We prove strong results about the laminar structure of survivable networks, which are outlined in Section 2. These results are of independent interest, and give useful techniques for dealing with survivable networks in both game theoretic and more traditional contexts.

Related Work. Over the last few years, there have been several new papers about the Connection Game, e.g., [2,8,10,12,14,15]. Recently, Hoefer [13] proved some interesting results for a generalization of the game in [4], and showed an interesting relationship between the Connection Game and Facility Location. While the survivable network design games that we consider can be expressed as part of the framework in [13], these results do not imply ours, and our results cannot be obtained using their techniques.

The research on non-cooperative network design and formation games is too much to survey here, see [16,18,21,23,24] and the references therein. Fabrikant et al. [11] (see also [1]) studied the price of anarchy of a very different network design game, and [5] considered the price of stability of a network design game with local interactions, intended to model the contracts made by Autonomous Systems in the Internet.

A major part of the research on network games has focused on congestion games [3,7,9,20,23]. Probably the most relevant such model to our research is presented in [3] (and further addressed in [7,8,12]). In [3], extra restrictions of "fair sharing" are added to the Connection Game, making it a congestion game and thereby guaranteeing some nice properties, like the existence of Nash equilibria and a bounded price of stability. While the Connection Game is not a congestion game, and is not guaranteed to have a Nash equilibrium, it actually behaves much better than [3] when all the agents are trying to connect to a single common node. Specifically, the price of stability in that case is 1, while the best known bound for the model in [3] is $\frac{\log n}{\log \log n}$ [19]. Moreover, all such models restrict the interactions of the agents to improve the quality of the outcomes, by forcing them to share the costs of edges in a particular way. This does not address the contexts when we are not allowed to place such restrictions on the agents, as would be the case when the agents are building the network together without some overseeing authority. However, as [4] has shown for the Connection Game and we show for the Survivable version of it, it is still possible to nudge the agents into an optimal outcome without restricting their behavior.

2 The Model and Structural Properties of OPT

We now formally define the Survivable Connection Game for N players. Let an undirected graph $G = (V, E)$ be given, with each edge e having a nonnegative cost $c(e)$, and let $s \in V$ be a special *root* (or source) node. Each player i has a single terminal node (also called *player node*) that she must connect to s using 2 edge-disjoint paths. The terminals of different players do not have to be distinct.

A strategy of a player is a payment function p_i, where $p_i(e)$ is how much player i is offering to contribute to the cost of edge e. Observe that players can share the cost of the edges. An edge e is considered *bought* if $\sum_i p_i(e) \geq c(e)$. Let G_p denote the subgraph of bought edges corresponding to the strategy vector $p = (p_1, \ldots, p_N)$. While strictly required to connect her terminals using at least 2 edge-disjoint connections, each player also tries to minimize her total payments, $\sum_{e \in E} p_i(e)$. Specifically, $\text{cost}(i) = \sum_{e \in E} p_i(e)$ if there are at least 2 edge-disjoint paths between the terminal of player i and s, and $\text{cost}(i) = \infty$ otherwise.

Assume G_p is the socially optimal solution OPT. Then, for every edge e, there is a set of players whose connection requirement will be dissatisfied if e is deleted from OPT, since otherwise a feasible network cheaper than OPT can be obtained by simply deleting e. Note that in a Nash equilibrium, only this set of players can make payments on e, since all other players will deviate by setting their payment on e to 0 if they have paid for e. The players in this set are therefore said to *witness* e, since without them, e would not be needed.

Let v be a player witnessing e, i.e., v will have only 1 path to s if e is deleted. Observe that the size of the min-cut between v and s in OPT is at least 2 and it becomes 1 when e is deleted. Therefore, there is a cut (A, B) in OPT between v and s of size 2 with e as one of the cut-edges. We call such a set of nodes A, a *witness set* of e. The 2 cut edges are called the boundary edges of the witness set since one side of them is in the set and the other side is outside the set. Note that since every edge in OPT has necessarily a witnessing player, it has a witness set as well, which can be constructed by the cut argument above. Figure 1(A) shows various witness sets of an edge e. The black circles represent the player nodes.

Definition 1. *A* witness set *of an edge e is a set of nodes including at least one player node and excluding s, with exactly 2 boundary edges, one of which is e.*

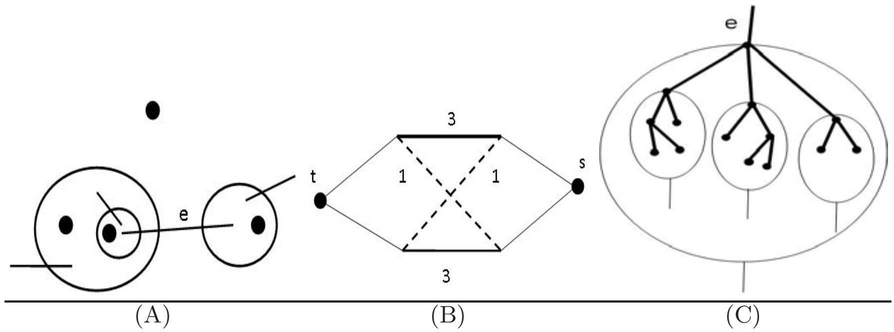


Fig. 1. (A) Shows various witness sets of an edge e. (B) An example illustrating our stable solution concept. (C) Result of the Tree Generation algorithm. The ovals represent smallest witness sets.

Observe that any player in a witness set of e witnesses e, and any player witnessing e has to be involved in some witness set of e. Intuitively, a player inside a witness set must use both of the edges leaving it, since the witness set is a cut of size 2 and she needs 2 disjoint paths. A player witnessing $e = (i, j)$ may be using e either in the direction $i \rightarrow j$ or in the direction $j \rightarrow i$. If it is using e in the direction $i \rightarrow j$ then it is inside a witness set containing i and it is inside a witness set containing j otherwise. Among the sets witnessing e in the direction $i \rightarrow j$, the smallest one in terms of the number of nodes included is called the *smallest witness set* of e in the direction $i \rightarrow j$ and we denote it by $W_i(i, j)$. $W_j(i, j)$ is also defined similarly. Smallest witness sets of the edges of OPT have very nice structural properties that we use in our proofs. Specifically, we rely on the following theorem. The proof of this theorem, as well as all of our missing proofs appear in the full version of the paper.

Theorem 1. *Let W be the set of all smallest witness sets of OPT, i.e., $W = \{W_i(e)|$ for some $i, e\}$. Then, there exists an equivalent graph (i.e., with the same price of stability) where W is laminar.*

Because of this theorem, for the rest of this paper we will assume that W is a *laminar* set system. In other words, for any two smallest witness sets W_1 and W_2 (they may be the smallest witness sets of different edges), either one of them is a subset of the other or they are disjoint. Because of this, we can now speak of $W_i(i, j)$ as the unique smallest witness set in the direction of $i \rightarrow j$, and therefore e may have 2 smallest witness sets, one in each direction. In the rest of this paper, we show how to construct a stable solution where only the players in the smallest witness sets of an edge e contribute to the payment of e. In fact, this laminar property holds not only for the optimal network, but also for any minimal feasible network G', i.e., where $G' - e$ is not feasible for any $e \in E(G')$. Therefore, if we do not possess OPT but some minimal feasible network, our techniques can still be used to obtain approximate equilibria with provable cost guarantees.

3 When All Nodes Are Terminals

In this section, we study a special case of the Survivable Connection Game, where each node of G is a player node, and prove that there always exists a Nash equilibrium as cheap as OPT, i.e., the price of stability is 1. To prove this result, we form a strategy profile p that buys the edges of OPT, i.e., we give an algorithm that decides how the cost of each edge of OPT is shared among the players. For each edge e of OPT, our algorithm only asks the adjacent terminals to contribute to the cost of e. Since we are trying to form a Nash equilibrium, each terminal can only contribute to the cost of the edges it witnesses. Though a terminal can have an arbitrary number of incident edges, it witnesses at most 2 of them. To see this, assume there is a player that witnesses more than 2 of its incident edges. Then at least one of the connection paths of this terminal is using at least 2 of its incident edges by the pigeonhole principle, which implies this connection path contains a cycle. Since that terminal is still 2-connected after removal of the cycle, it does not witness the incident edges included in the cycle. That observation gives us a nice substructure in OPT, which we call *chains*.

A *chain* is a path with maximal length in OPT, where each edge of the path has 2 smallest witness sets. Observe that each intermediate node of the chain is witnessing both of its incident edges in the chain, since every edge has 2 smallest witness sets, one containing each of its incident nodes. Since a terminal can witness at most 2 of its incident edges, no intermediate node of the chain witnesses any incident edge except the ones in the chain. The boundary nodes of the chain are witnessing the edge of the chain they are adjacent to. Observe that boundary nodes of the chain may or may not witness any other incident edge but if they do, this incident edge they witness will have only 1 smallest witness set, since otherwise this edge would have been part of the chain as well.

Observe that every edge e with 2 smallest witness sets is included in some chain. In the simplest case, where both of the adjacent nodes of e do not witness any other incident edges or witness one other edge with 1 smallest witness set, we will have a chain that includes only one edge, namely e. Therefore, OPT is composed of chains and edges with 1 smallest witness set. To form the stable solution, we first form the payment on the edges with only 1 smallest witness set, and then form the payments on the edges of the chains.

Since we are trying to form a stable solution, we should never ask the players to make a payment that will create an incentive of unilateral deviation. To ensure this, whenever we ask a player i to contribute to the cost of an edge e, the algorithm should compute the cost of the cheapest deviation χ_i for player i on the edges of $G-e$. Observe that all edges of OPT such that i is not contributing any payment to them can be used by i freely. Therefore, when computing χ_i, the algorithm should not use the actual cost of the edges in $G - e$, but instead for each edge f it should use the cost i would face if she is to use f. We call this the *modified cost of f for i*, and denote it by $c'(f)$. Specifically, for f not in OPT, $c'(f) = c(f)$, f's actual cost, since this is how much i would have to pay to purchase f. If i is already paying some amount $p_i(f)$ for f, then $c'(f) = p_i(f)$, since i has to continue paying this amount to use f. And finally, if i is paying nothing for f that is in OPT (or has not been asked to pay anything for it yet by our algorithm), then $c'(f) = 0$. Therefore, from i's perspective, all the edges of OPT that are not adjacent to it, or that it is not witnessing, are always free, since our payment scheme will never ask i to pay for them.

Payment Algorithm. The algorithm first loops through all edges of OPT that have only 1 smallest witness set. Let $e = (i, j)$ be one of those edges and without loss of generality assume i is the witnessing adjacent player. Then the algorithm asks i to pay for the whole cost of e. As mentioned above, it also computes the cost of the cheapest deviation χ_i. If $\chi_i \geq \sum_{j \neq e} p_i(j) + c(e)$ then it sets $p_i(e) = c(e)$ and proceed with the next edge. If $\chi_i < \sum_{j \neq e} p_j(e) + c(e)$ then the algorithm breaks (we prove below that this can never happen). If the algorithm succeeds in paying for all the edges with 1 smallest witness set, i.e., it does not break, then we consider the payment of the chains. We loop through all the chains C of OPT. Let $e_1 = (n_1, n_2), e_2 = (n_2, n_3), \ldots, e_k = (n_k, n_{k+1})$ be the edges of a chain C. To form the payment on the edges of C, the algorithm loops through all the edges of C starting from the leftmost edge e_1 till the rightmost edge e_k. So the payment for e_i is decided after the payments for $e_1, e_2, \ldots, e_{i-1}$ are already decided. To form the payment for e_i, the algorithm asks n_i to make the maximum payment that will not create an incentive of unilateral deviation. The algorithm then asks n_{i+1} to pay for the rest of the cost of e_i while not creating an incentive for unilateral deviation. If the adjacent nodes succeed in paying for the edge, the algorithm continues with the next edge of the chain. Otherwise, the algorithm breaks. If the algorithm succeeds in paying for the chain, it proceed to the next chain.

Theorem 2. *The payment algorithm forms a Nash equilibrium on the edges of OPT, and therefore the price of stability is* 1.

Proof. [Summary] Observe that the payment algorithm never asks a player to make a payment that will create an incentive of unilateral deviation. Therefore, in order to show the price of stability is 1, all we need to do is to prove that the algorithm never breaks at an intermediate stage. We prove this by constructing a feasible network cheaper than OPT whenever the algorithm breaks, which will contradict the optimality of OPT. We first show that players will always pay for edges $e = (i, j)$ that have only one smallest witness set. If the player i does not pay for e, then we can make player i deviate, i.e., play χ_i instead of p_i. The new network is cheaper than OPT and we prove that it is also a feasible network by showing all the players that were witnessing e can "follow" i's deviation. We then proceed to show that chains are always paid for. If a chain $(n_1, n_2, \ldots, n_k)$ cannot be paid for, we form a network cheaper than OPT by letting a contiguous subset of players $n_1, n_2, \ldots, n_k$ deviate. Because of the specific order in which we form the deviations, the new network is still feasible by similar but slightly more complicated arguments. □

4 Good Stable Solutions When Not All Nodes Are Terminals

Approximation Algorithm Technique. To prove our main result, we define a restricted version of the Survivable Connection Game such that this version of the game is identical to the original game, except that for each strategy p_i of a player i, the set of deviations she can take are restricted. In this version, a player is only allowed to deviate by changing the payments on one of her paths, instead of both of them at once, i.e., for any strategy profile $p = (p_1, \ldots, p_n)$, p'_i is a deviation for player i from her strategy p_i if for each edge e along *one* path from i to s in G_p, $p_i(e) = p'_i(e)$. In this version of the game, each player should also determine her connection paths as well as the payment she makes on the edges as part of her strategy. To avoid ambiguity, in the rest of the paper we will use the term *stable solutions* for the equilibria of the restricted version and the results for the stability of the restricted version will imply results about the Nash equilibria of the Survivable Connection game due to Theorem 3.

Theorem 3. *A stable solution p is a 2-approximate Nash equilibrium of the original Survivable Connection Game.*

The restricted version of the game is also of independent interest in scenarios where each path of a single player is managed by a different entity.

In Figure 1(B), we have a game with one player that wants to connect from t to s through 2 edge-disjoint paths. Each thick edge has a cost of 3, each dashed edge has a cost of 1 and the total cost of the thin edges is ϵ. Any feasible solution has to include all 4 of the thin edges. Let p be a strategy of the player where she buys the 2 thick edges and all 4 of the thin edges where she uses the upper path and the lower path in Figure 1(B) as her connection paths. Please note that, though the connection paths are uniquely determined on this game for this set of bought edges, this is not true in general, therefore they are to be specified as part of the strategy. Let p' be a strategy where the player buys the dashed and the thin edges as well as the top thick edge and for each connection path she uses a dashed edge and its 2 incident thin edges. Observe that in this strategy player buys the top thick edge although she does not use it. If the player switches her strategy from p to p', she reroutes both of her connection paths. However, p' is considered a valid deviation since she keeps the payments on one of her connection paths in p the same.

Recall that because of Theorem 3, we can restrict our attention to *stable* solutions as defined above. In the following discussion we will use the terms *price of anarchy* and *price of stability* for the ratio of the cost of the worst and best *stable* solutions to the cost of OPT. In fact, we can immediately observe that the price of anarchy cannot be more than $2N$, and we give an example in the full version that shows this bound is tight. Because of this, we focus on the price of stability.

In this section, we present an algorithm to find a stable strategy vector that buys OPT, which implies that the price of stability for the *Survivable Connection*

Game is 1. Since a strategy of a player is composed of specifying 2 edge-disjoint paths to s and the amount of payment made on the edges, then we must specify both of these for every player.

Theorem 4 states that there always exists an *equivalent graph* G' where each player has node-disjoint connection paths on the socially optimal solution. Equivalence among the graphs means that the socially optimal solution of the new graph costs as much as OPT, and that for each stable network in the new graph, there corresponds a unique stable network in the original graph with the same cost. Because of Theorem 4, we assume all players have a node-disjoint routing on OPT in the rest of the paper. We fix an arbitrary such routing for all the players and next show how to form payments on the edges of OPT.

Theorem 4. *There exists an equivalent graph G' with a routing on the socially optimal solution that is node-disjoint.*

Our payment scheme is formed by Algorithm 1. While deciding the payment on an edge $e = (u, v)$, the algorithm needs to form the cheapest deviation χ_i on $G - e$, for all players i in $W_u(u, v)$ and $W_v(u, v)$. For each player i in $W_u(u, v)$ or $W_v(u, v)$, we call the connection path of i that does not use e the *enduring path* of player i and denote it as E_i. To form the cheapest deviation χ_i in this algorithm, we need to be able to find the cheapest way for a player to form 2 edge-disjoint paths to s, while keeping the payments on E_i the same. As shown in Algorithm 1, this can be done by using modified costs $c'_i(f)$ for each edge f, that represent how much it costs for player i to use edge f in χ_i. Specifically, for f not in OPT, $c'_i(f) = c(f)$, the actual cost of f. For the edges f of OPT that i has not paid anything for, or for the edges in E_i, we have that $c'_i(f) = 0$, since from i's perspective, she can use these edges for free (she cannot change the payments on E_i, so from a deviational point of view, those edges are free for i to use in χ_i). For all the other edges f that i is paying $p_i(f)$ for, $c'_i(f) = p_i(f)$, since that is how much it costs for i to use f in her deviation χ_i.

Note that Algorithm 1 never asks a player i to pay more than the cost of her cheapest deviation χ_i and so, the algorithm forms a stable solution if it

```
Initialize p_i(e) = 0 for all players i and edges e;
Loop until the payments for all edges are determined;
    Pick an edge e = (u,v) whose payment scheme hasn't been decided;
    Pay for all the edges in e's smallest witness sets recursively;
    Loop through all terminals i of W_u(u,v) and W_v(u,v) until e is
    paid for;
        Define p_i = Σ_{f∈(E\E_i)} p_i(f);
        Define p(e) = Σ_j p_j(e);
        Define c'_i(f) to be the modified cost of f for i;
        Define χ_i to be the cost of the cheapest deviation by player
        i in G − e under c'_i;
        Set p_i(e) = min{χ_i − p_i, c(e) − p(e)}.
```

Algorithm 1. Algorithm That Generates the Payment Scheme terminates. Therefore, all we need to prove is that for any edge e, the terminals inside its smallest witness sets will be willing to pay for e. To show this, we will actually prove a stronger statement. Specifically, for every edge $e = (i, j)$ we will generate two trees T_i and T_j in $W_i(e)$ and $W_j(e)$ rooted at i and j respectively, such that the leaves of T_i and T_j are player nodes/terminals, and all other nodes are non-player nodes. We will show that just the leaves of these trees are willing to pay for the whole cost of e, and the other terminals in the smallest witness sets are not needed. In fact, we can just as easily make our algorithm only ask the players that are leaves of these trees to contribute to the payment of e.

Tree Generation. Since W is laminar, we construct the trees $T_j(i, j)$ recursively, starting with the smallest sets in W, and continuing to the sets containing those. To construct $T_j(i, j)$, we start the search in $W_j(i, j)$ from j. If j is a player then the tree is just a single node. If it is a non-player node we add all its incident edges in $W_j(i, j)$, along with their corresponding trees in their smallest witness sets from the other side, as shown in Figure 1(C). That is, for every edge (j, k) inside $W_j(i, j)$, we add the edge (j, k) and the subtree $T_k(j, k)$. These trees must have already been generated, since those witness sets are contained inside $W_j(i, j)$. We presented the tree generation in terms of the smallest witness sets but indeed it is equivalent to making a breadth-first search in $W_j(i, j)$ starting from j, except we stop when a branch arrives at a player node. The following lemma proves that the tree generation algorithm is well-defined and the structures it generates $T_i(e)$ and $T_j(e)$ are indeed trees.

Lemma 1. *Any edge f of $T_i(e)$ generated by the Tree Generation Algorithm has a smallest witness set from the side of the lower level nodes of the tree. Furthermore, the structure $T_j(e)$ generated by the Tree Generation Algorithm for any edge $e = (i, j)$ of OPT is a tree such that all leaf-nodes are player nodes and all non-leaf nodes are non-player nodes.*

We now know each player node t at a leaf of a tree $T_j(e)$ is in the smallest witness set of all the edges of the path of the tree between her and j. This implies that every one of these edges *must* be used by t to connect to s, and since the connection paths to s are node-disjoint, this implies that one of the connection paths of t must simply proceed up the tree $T_j(e)$. Therefore, we know that the other connection path of t does not use any edge of this path. Lemma 2, which is one of the key lemmas for our proof, shows an even stronger property and states that the connection paths of all players leaving $W_j(i, j)$ through the other boundary edge (i.e., not the edge (i, j)) don't use any edge of $T_j(e)$ at all.

Lemma 2. *Let $W_j(i, j)$ be a smallest witness set of some arbitrary edge $e = (i, j)$. Let p be a player inside $W_j(i, j)$. Then the other connection path of p (that leaves $W_j(i, j)$ through the other boundary edge) does not use any of the edges of $T_j(e)$.*

Now that we generated the trees $T_i(e)$ inside each smallest witness set that are disjoint from the other connection paths of the player nodes, we are ready to state our main theorem.

Theorem 5. *Algorithm 1 fully pays for OPT, and so the price of stability is* 1*. Moreover, the leaves of $T_i(e)$ and $T_j(e)$ are willing to pay for an edge $e = (i, j)$ without help from any other players.*

Proof. [Summary] To prove Algorithm 1 succeeds in paying for OPT, we show that for any edge e, the leaf-nodes of $T_i(i, j)$ and $T_j(i, j)$ will be willing to pay for e. Assume the players are unable to buy an edge e. Then each player at the leaf of its corresponding trees has some deviation which explains why she cannot contribute more to the cost of e. As before, our proof is based on obtaining a feasible network cheaper than OPT by modifying OPT with a careful subset of these paths.

Recall the deviation χ_k found in Algorithm 1, which consisted of edge-disjoint paths P_1 and P_2 from a terminal t_k to s, together with t_k's enduring path. Define player t_k's *alternate connection cycle* A_k to be the paths $P_1 \cup P_2$. If there is more than one such deviation, choose A_k to be the one which includes as many ancestors of t_k as possible before including edges outside the tree. Let d_k be the highest ancestor of t_k that A_k reaches before leaving the tree. To show that all edges are paid for, we need the following technical lemma concerning the structure of the alternate connection cycles.

Lemma 3. Let player t_k be a leaf-node of $T_i(i, j)$. Then A_k, the alternate connection cycle of t_k, does not use any edge of $T_i(i, j)$ except in the subtree below d_k.

Using the above lemma, we can now prove Theorem 5. If the leaves of the trees $T_i(i, j)$ and $T_j(i, j)$ together cannot pay for the edge (i, j), then we let a certain subset of these players deviate. Afterwards, we must show that the resulting solution is still feasible, by showing that everyone still has 2 edge-disjoint paths to s. The complications that arise here result from the fact that a terminal t cannot simply "follow" the deviating paths of a terminal t', since the deviation of t' might not be disjoint from the other connection path of t. We can still show, however, that the resulting solution is feasible and cheaper than OPT, giving us a contradiction.

Since computing OPT is computationally infeasible, we give the following result.

Theorem 6. *Given an α-approximate socially optimal graph G_α and any $\epsilon > 0$, there is a polynomial time algorithm which returns a $2(1 + \epsilon)$-approximate Nash equilibrium on a feasible graph G', where $c(G') \leq c(G_\alpha)$.*

References

1. Albers, S., Eilts, S., Even-Dar, E., Mansour, Y., Roditty, L.: On Nash Equilibria for a Network Creation Game. In: SODA (2006)
2. Anshelevich, E., Caskurlu, B.: Exact and Approximate Equilibria for Optimal Group Network Formation. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 239–250. Springer, Heidelberg (2009)

3. Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, É., Wexler, T., Roughgarden, T.: The Price of Stability for Network Design with Fair Cost Allocation. In: Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (2004)
4. Anshelevich, E., Dasgupta, A., Tardos, É., Wexler, T.: Near-Optimal Network Design with Selfish Agents. In: Proc. 35th ACM Symposium on Theory of Computing (2003)
5. Anshelevich, E., Shepherd, B., Wilfong, G.: Strategic Network Formation through Peering and Service Agreements. In: FOCS 2006 (2006)
6. Bala, V., Goyal, S.: A Non-Cooperative Model of Network Formation. Econometrica 68, 1181–1229 (2000)
7. Chekuri, C., Chuzhoy, J., Lewin-Eytan, L., Naor, J., Orda, A.: Non-cooperative multicast and facility location games. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC), Ann Arbor, Michigan, pp. 72–81 (2006)
8. Chen, H., Roughgarden, T.: Network Design with Weighted Players. In: SPAA 2006 (2006)
9. Christodoulou, G., Koutsoupias, E.: On the price of anarchy and stability of correlated equilibria of linear congestion games. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 59–70. Springer, Heidelberg (2005)
10. Epstein, A., Feldman, M., Mansour, Y.: Strong Equilibrium in Cost-Sharing Connection Games. In: EC 2007 (2007)
11. Fabrikant, A., Luthra, A., Maneva, E., Papadimitriou, S., Shenker, S.: On a Network Creation Game. In: PODC (2003)
12. Fiat, A., Kaplan, H., Levy, M., Olonetsky, S., Shabo, R.: On the price of stability for designing undirected networks with fair cost allocations. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 608–618. Springer, Heidelberg (2006)
13. Hoefer, M.: Non-cooperative facility location and covering games. In: Asano, T. (ed.) ISAAC 2006. LNCS, vol. 4288, pp. 369–378. Springer, Heidelberg (2006)
14. Hoefer, M.: Non-cooperative Tree Creation. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 517–527. Springer, Heidelberg (2006)
15. Hoefer, M., Krysta, P.: Geometric network design with selfish agents. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 167–178. Springer, Heidelberg (2005)
16. Jackson, M.: A survey of models of network formation: stability and efficiency. In: Demange, G., Wooders, M. (eds.) Group Formation in Economics: Networks, Clubs and Coalitions. Cambridge Univ. Press, Cambridge
17. Jain, K.: A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. Combinatorica 21 (2001)
18. Johari, R., Tsitsiklis, J.: Efficiency loss in a network resource allocation game. Mathematics of Operations Research 29(3), 407–435
19. Li, J.: An $O\left(\frac{logn}{loglogn}\right)$ upper bound on the price of stability for undirected Shapley network design games. Information Processing Letters 109(15), 876–878
20. Monderer, D., Shapley, L.: Potential Games. Games and Economic Behavior 14, 124–143 (1996)
21. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V. (eds.): Algorithmic Game Theory. Cambridge University Press, Cambridge
22. Rosenthal, R.W.: The network equilibrium problem in integers. Networks 3, 53–59 (1973)
23. Roughgarden, T.: Selfish Routing and the Price of Anarchy. MIT Press, Cambridge
24. Tardos, É.: Network Games. In: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing (2004)

Games with Congestion-Averse Utilities

Andrew Byde[1], Maria Polukarov[2], and Nicholas R. Jennings[2]

[1] Hewlett-Packard Laboratories, Bristol, UK
andrew.byde@hp.com
[2] School of Electronics and Computer Science,
University of Southampton, UK
{mp3,nrj}@ecs.soton.ac.uk

Abstract. Congestion games—in which players strategically choose from a set of "resources" and derive utilities that depend on the congestion on each resource—are important in a wide range of applications. However, to date, such games have been constrained to use utility functions that are linear sums with respect to resources. To remove this restriction, this paper provides a significant generalisation to the case where a player's payoff can be given by *any* real-valued function over the set of possible congestion vectors. Under reasonable assumptions on the structure of player strategy spaces, we constructively prove the existence of a pure strategy equilibrium for the very wide class of these generalised games in which player utility functions are *congestion-averse*—i.e., monotonic, submodular and independent of irrelevant alternatives. Although, as we show, these games do not admit a generalised ordinal potential function (and hence—the finite improvement property), any such game does possess a Nash equilibrium in pure strategies. A polynomial time algorithm for computing such an equilibrium is presented.

1 Introduction

Models of congestion have recently become a major issue of study in algorithmic game theory, as they arise from many real-life situations (examples include network routing, resource and task allocation, competition of firms for production processes [8,16,17]) and yet possess plausible theoretical properties. To date, much of research deals with the model of *congestion games* introduced by Rosenthal [16]. Here, players share a finite set of resources, and a strategy for a player is to choose a subset of the resources. Each resource is associated with a resource utility function, which determines the utility of each of its users as a function of the number of players that have selected the resource. Given a strategy profile, a combination of the players' chosen strategies, the payoff for a player will be simply the sum of utilities from his utilised resources.

Now, the key result of Rosenthal is that congestion games always possess pure strategy Nash equilibria. This is important because pure strategy equilibria have some indisputable advantages over mixed strategy equilibria: they are more intuitive, especially in the context of one-shot games, they are generally easier to compute than mixed equilibria, and they are easier for players to coordinate to.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 220–232, 2009.

However, there are only a few known classes of such games with pure equilibria, and, to date, there has been relatively little work providing efficient and exact algorithms for computing such equilibria.

Related Work and Motivation. To this end, Monderer and Shapley [10] introduced the notion of *potential function* (and its relaxed version—*(generalised) ordinal potential* function) and proved that the existence of a generalised ordinal potential is equivalent to the *finite improvement property* (FIP), implying that any sequence of unilateral improving deviations terminates at a pure strategy Nash equilibrium. The authors also showed that the classes of finite—exact—potential games and congestion games coincide.

More recently, congestion games have been extended to *local-effect games* [6], *player-specific congestion games* [7], *weighted congestion games* [7] and *ID-congestion games* [9]. In these models, the player's payoff depends not only on the number of players choosing his resources, but also on the number of players choosing the neighboring resources or on the players' identities. Additional generalizations [11,15,13,14] deal with the possibility that resources may fail to execute their assigned tasks, or with the actual order in which the tasks are executed; we will refer to these models as *congestion games with faulty or random-order services*. Finally, much of the work has been devoted to the study of the computational complexity of finding pure strategy equilibria [1,5] and their social performance [2,3] in congestion games and some of the extended models.

In this paper we generalise the class of congestion games still further. Specifically, we consider settings in which the payoff of a player is determined by the vector of resource congestion (note that the resources might be mutually dependent!), via any real-valued function—not just as a sum of resource-specific functions. Clearly, in this very general setting a potential function and a pure strategy equilibrium are not guaranteed to exist. However, under reasonable assumptions on the structure of player strategy spaces and payoff functions, we will prove the existence of a pure strategy equilibrium and develop a polynomial time algorithm for the computation of such an equilibrium.

Specifically, we assume that each player has a set of accessible resources, which is a subset of a given set of resources to share, and his strategy space consists of all possible subsets of his set of resources at hand. This, for instance, captures settings with typed resources, where subsets of resources of a particular type are matched to particular player tasks, or situations where some resources are not (physically) accessible by a particular player or the player's permissions for the resource use are restricted.

In many applications of congestion games discussed in the literature, the resource utility function is *decreasing* as a function of the number of users (or, the resource cost function is increasing). This typically reflects situations where a resource is a service provider whose costs per user are increasing due to competition on internal resources, or a player's utility from a resource decreases due to reduction in the resource efficiency or reliability caused by higher congestion. The latter also gives rise to the very real issue of redundant usage of resources, which often occurs in non-cooperative multi-agent systems, where selfish agents try to run their jobs on several resources in parallel, in an attempt to increase the probability of success or in the hope that one will be faster. In such settings it is natural to assume that the more resources a player has in use or the lower is the congestion on his utilised resources, the less marginal benefit he derives from hiring an additional resource—that is, the player's utility is *submodular.* This is also the case in multi-task allocation settings with concave values of task portfolios and other scenarios. Finally, a player's preference between two resources is usually determined by the congestion these resources experience but is *independent of irrelevant alternatives*—that is, of selection and congestion levels of the other resources.

Contribution. Following the above motivation, we introduce the class of *games with congestion-averse utilities* (CAGs) where the structure of strategy spaces is the one described above, and the payoff functions are *congestion-averse*—that is, monotonically decreasing, submodular and independent of irrelevant alternatives. We observe that the presented class of games includes—but is not restricted to—the earlier discussed congestion models with player-specific payoff functions, or faulty/random-order services. Indeed, CAGs significantly generalise the aforementioned models as they, in particular:

– allow for more general—and non-identical—player payoff function structures. Informally, when utilities and costs are understood metaphorically (e.g. as a monetisation of a benefit or inconvenience) this allows us to model different degrees of motivation/impatience between players; when they are understood literally it allows us to model utility/cost differentiation on behalf of the resource provider;
– take into account the possibility that players may have unequal access to different resources. This allows us to model player-specific tasks which can only be executed by a certain collection of resources;
– allow for non-identical and mutually-dependent resources;
– can be used to model multi-task allocation and other complex scenarios.

The main results of this paper are as follows. We observe that CAGs do not, in general, admit a generalised ordinal potential function and the finite improvement property. However, we prove that every such game possesses a Nash equilibrium in pure strategies and that any strategy profile which is stable under elementary changes (adds, drops or switches of a single resource), is a Nash equilibirum—this is called *the single profitable move property* (SPMP). Moreover, we show that the family of games with SPMP coincides with the class of CAGs, and thus our result is complete. Finally, we develop a universal, polynomial time algorithm that computes a pure strategy equilibrium in any given CAG, while the methods previously developed for models with player-specific payoff functions or faulty/random-order services, which are special cases of CAGs, appear to fail in the general case. Our new technique is based on the special sequences of elementary changes which we call "drop ladders" and "swap ladders". In particular, we show that an equilibrium can be achieved by applying $O(N^2R^2)$

of elementary changes, where N and R stand for the number of players and resources, respectively. Most of the proofs are omitted, due to space limitations.

2 Games with Congestion-Averse Utilities

Consider a congestion setting with a set $\mathbf{N} = \{1, \ldots, N\}$ of players, where each player $i \in \mathbf{N}$ has a set $\mathbf{R}_i$ of $R_i \in \mathbb{N}$ accessible resources, which is a subset of a finite superset $\mathbf{R} = \{r_1, \ldots, r_R\}$. A player i's strategy is to choose a subset of resources from $\mathbf{R}_i$, i.e. i's strategy space, Σ_i, is given by a power set of $\mathbf{R}_i$ (either including the empty set or not, depending on the nature of the particular application). We refer to a resource that a player has assigned a task to as "selected" by that player, and "unselected" otherwise.

Every N-tuple of strategies $\sigma = (\sigma_i)_{i \in \mathbf{N}}$ corresponds to an R-dimensional congestion vector $h(\sigma) = (h_r(\sigma))_{r \in \mathbf{R}}$ where $h_r(\sigma)$ is the number of players who select resource r (we drop the profile to give h_r when it's clear which profile is under consideration). Given a strategy profile $\sigma \in \Sigma$, for any player $i \in \mathbf{N}$, we define his *personalised* vector of congestion, $h^i(\sigma)$, to be a vector in $\mathbb{N}^R$ that coincides with $h(\sigma)$ for all the resources that have been selected by i and that has zero entries for all of i's unselected resources: that is, $h^i_r(\sigma) = h_r(\sigma)$ if $r \in \sigma_i$ and $h^i_r(\sigma) = 0$ otherwise. For a vector $h \in \mathbb{N}^R$ we define the "support" of h, $S(h) \subseteq \{1, \ldots, R\}$, to be $\{j : h_{r_j} > 0\}$.

In a classic congestion game [16], the payoff function of player i is defined by $U_i(\sigma) = \sum_{r \in \sigma_i} u_r\left(h_r(\sigma)\right)$, where $u_r : \{1, \ldots, N\} \to \mathbb{R}$, $r \in \mathbf{R}$, is an assignment of resource utility functions; for any resource $r \in \mathbf{R}$, $u_r(k)$ denotes the utility for a player from using resource r if the total number of users of r is k.

In our, generalised, model, the utility of player i in a congestion setting is given by a function $U_i : \mathbb{N}^R \to \mathbb{R}$ that assigns a real value to a **vector** of congestion.[1] To precisely define the set of utility functions we permit, it is first necessary to introduce a set of strategy modifications which we call "elementary changes"[2]. Given a profile $\sigma \in \Sigma$, we denote the elementary changes as follows:

- **Add** $A_i(r)$—player i *adds* an unselected resource r: $\sigma'_i = \sigma_i \cup \{r\}$.
- **Drop** $D_i(r)$—player i *drops* a selected resource r: $\sigma'_i = \sigma_i \setminus \{r\}$.
- **Switch** $S_i(r_+ \leftarrow r_-)$—player i *switches* resources by adding resource r_+ and dropping resource r_- (note that $S_i(r_+ \leftarrow r_-) = A_i(r_+) + D_i(r_-)$[3]

Using the above notation, we now define the "congestion-averseness" conditions on the players' utilities. Here, a utility function is said to be congestion-averse if it (i) monotonically decreases as congestion increases, (ii) is submodular in

[1] Note that the player's utility only depends on the numbers of players choosing each resource but not on their identities—that is, this setting is *anonymous* (see [4] for results on approximating equilibria in anonymous games).

[2] In [12,13] they are referred to as "single moves".

[3] Here and in what follows, "+" should be understood to mean sequential execution, read left-to-right. We also use this notation to indicate elementary changes applied to strategy profiles: e.g., $\sigma + D$ denotes a drop applied to profile σ.

that the "better" collection of resources a player uses—the less incentive he has to add new resources, and (iii) is independent of irrelevant alternatives (i.e., a player's preference between two resources depends only on congestion on the resources in question). Formally,

Definition 1. *A utility function $U : \mathbb{N}^R \to \mathbb{R}$ is described as* ***congestion-averse*** *if it satisfies the following three conditions:*

- ***Monotonicity*** (M). *Function U is monotonically decreasing with respect to increasing congestion: if $S(h) = S(h')$ and $\forall r$, $h_r \geq h'_r$, then $U(h) \leq U(h')$.*
- ***Submodularity*** (SM). *Improving a resource selection by either (i) profitable switches, (ii) extending the set of utilised resources or (iii) reducing congestion on them does not make new adds more profitable, or drops less profitable; likewise, unprofitable switches, deleting the resources or increasing the congestion does not make drops more profitable, or adds less profitable. Equivalently, for any h, h' and h'' such that $|S(h)| = 1$ and $S(h) \nsubseteq S(h'), S(h'')$,*

$$U(h + h') - U(h') \leq U(h + h'') - U(h''),$$

  if either (i) $|S(h') \setminus S(h'')| = |S(h'') \setminus S(h')| = 1$ and $U(h') \geq U(h'')$, (ii) $S(h'') \subseteq S(h')$ and $h_j{}'' = h_j{}'$ for any $j \in S(h'')$, or (iii) $S(h') = S(h'')$ and $h' \leq h''$.
- ***Independence of irrelevant alternatives*** (IIA). *If a player "prefers" one resource over another at their current congestion levels, then he still does so no matter what other changes are made to any other resources. Formally, if $S_i(r_+ \leftarrow r_-)$ is a profitable switch for player i given profile σ, then it is profitable for i from any other profile σ' satisfying $r_- \in \sigma'_i$, $r_+ \notin \sigma'_i$, $h_{r_-}(\sigma) = h_{r_-}(\sigma')$ and $h_{r_+}(\sigma) = h_{r_+}(\sigma')$.*

A CAG is now defined as a game in the congestion domain with congestion-averse utility functions, in which a player's utility from a combination of strategies is determined by his personalised vector of congestion. More presicely,

Definition 2. *A* ***CAG*** *$\Gamma = \big(\mathbf{N}, \mathbf{R}, (U_i(\cdot))_{i \in \mathbf{N}}\big)$ consists of a set $\mathbf{N}$ of $N \in \mathbb{N}$ players, a set $\mathbf{R}$ of $R \in \mathbb{N}$ resources, and for each player i a set of accessible resources $\mathbf{R}_i \subseteq \mathbf{R}$ and a congestion-averse utility function $U_i : \mathbb{N}^R \to \mathbb{R}$. The strategy space for each player $i \in \mathbf{N}$ is the set of subsets of $\boldsymbol{R}_i$, and the payoff to the player from a combination of strategies σ is $u_i(\sigma) = U_i(h^i(\sigma))$, where $h^i(\sigma)$ is i's personalised vector of congestion as determined by σ.*

As we have previously discussed, congestion-averseness is a very reasonable assumption that is natural in many applications of congestion settings. In particular, we note that the independence of irrelevant alternatives holds for classic congestion games and **all** their known up to date extensions and generalisations. Interestingly, these are the only conditions we will need on the players' utility functions in order to guarantee a pure strategy equilibrium. Before we proceed to the proof, however, we point out some interesting subclasses of CAGs.

Interesting Subclasses. Although congestion games as a whole are unlikely to be included in the class of CAGs (as they are, in general, PLS-complete [5]), the presented model captures various scenarios that lie far beyond the borders of the classic model. Below we provide a number of examples of generalised congestion settings, previously studied in the literature, to illustrate the richness of our model. We note, however, that the class of CAGs is **not** restricted to these special cases, as we shall also learn from their properties discussed in the sequel.

Probably the first significant extension of congestion games was the work of Milchtaich on *congestion games with player-specific functions* [7], in which the utility function associated with each resource is not universal but player-specific. This generalisation was accompanied by two limiting assumptions: (i) that each player chooses only one resource, and (ii) that the utility he derives from a particular resource decreases with congestion on it. Now, one can observe that a player-specific congestion game can be easily modified to a CAG by assuming that a player can choose any (non-empty) subset of originally available single resources, and by setting the utility of a player i from a congestion vector h to be given by $U_i(h) = \sum_{r \in S(h)} u^i_r(h_r) - M(|h|-1)$, where $u^i_r(\cdot)$ represent resource utility functions in the original game and M is a sufficiently big number (say, $M > \sum_{r \in \mathbf{R}} \max_{i \in \mathbf{N}} u^i_r(1)$) (note that for a vector h with $S(h) = 1$ this coincides with the original resource utility function). Obviously, the sets of outcomes of these games are identical, and the congestion-averse conditions are satisfied.

Another interesting example is the family of congestion models with faulty or random-order services, that, specifically, includes *taxed congestion games with failures* [15], *congestion games with load-dependent failures* [13] and *random order congestion games*[4] [14]. In each of these, a player has a task that can be carried out by any element of a set of independent resources. A player may decide to assign his task, simultaneously, to several resources, either for reliability reasons or hoping that his task will be completed in a short time by at least one of the resources (all possible subsets of all given resources are available to all of the players). Doing this, each player wants to maximise the probability of successful (or, quick) completion of his task and, simultaneously, to minimise his cost. It has been (naturally) assumed that the failure probabilities increase with congestion, implying the monotonicity of player utility functions, and that the marginal benefit from hiring an additional resource decreases as the player's selection of resources improves, implying the submodularity. Finally, since resources are uncorrelated, the IIA also holds.[5]

We also note that CAGs can be used to model more complex—rather than "single job"—scenarios. Consider, e.g., a setting where a player is given different tasks, each associated with a value and workload. A subset of resources can complete a task if its productivity (which is a function of the number and congestion on the resources) meets the task's workload. A player strives to maximise the total value of his completed tasks, and thus he is interested in executing as many (valuable) tasks as possible.

[4] These games are also referred to as *asynchronous* congestion games [12].

[5] We omit the formal proofs and definitions for brevity of exposition.

Finally, as we discussed earlier in the Introduction, CAGs possess additional features allowing for modeling player-specific tasks, non-identical and mutually dependent resources. Thus, while the above are interesting models of specific task allocation problems, games with congestion-averse utilities provide a general framework for more realistic modelling of congestion scenarios.

3 Properties of CAGs

In this section we investigate the properties of games with congestion-averse utilities. In particular, we observe that these games do not admit a generalised ordinal potential function and the FIP. However, as we show below, they do possess the single profitable move property. Based on this, we develop our "drop and swap ladders" technique that enables us to achieve a pure strategy Nash equilibrium in any given CAG, while the algorithms previously developed for specific subclasses fail in the general case.

3.1 The Non-existence of the FIP

The finite improvement property is equivalent to the existence of a generalised ordinal potential function—a real-valued function over the set of pure strategy profiles with the property that an increase in the utility of a player who unilaterally shifts to another strategy implies an increase in the potential function; that is, the potential increases along any improvement path.

Based on this, we construct an example showing that CAGs, in general, have no FIP. In fact, this can be concluded directly from the inclusion in the class of CAGs of congestion games with player-specific payoff functions [7], for which examples of games without FIP have been previously found. In contrast, although the models with faulty/random-order services [11,15,13,14] have been shown to not admit an exact potential function, the previous work failed to prove or disprove the existence of an ordinal potential function in these games.

Note, however, that the absence of the FIP, in general, does not contradict the existence of an equilibrium in pure strategies or the convergence of particular one-sided better reply dynamics. In what follows, we consider special types of improving deviations and the corresponding properties of games with congestion-averse utilities that we will use to develop efficient procedures for constructing pure strategy equilibria.

3.2 The Single Profitable Move Property

The simplest deviations from a strategy profile in a CAG involve adds, drops or switches of single resources, referred to as *elementary changes* (see 2 for the formal definition). As we show below, any profile which is stable against elementary changes possesses no profitable deviations at all, and hence is a Nash equilibrium.

The above property, referred to as the *single profitable move property* (SPMP), has been previously shown to hold for special cases of games with faulty/random-order services [15,13,14].[6] Here we extend this result to the superclass of CAGs. We then complete the result by showing that the congestion-averse conditions are not just sufficient but also necessary for the existence of SPMP (see 3.4).

Theorem 1. *A strategy profile σ of a given CAG is a pure strategy Nash equilibrium if and only if it possesses no profitable elementary changes.*

This allows us to significantly reduce the size of the set of possible player deviations from a given strategy profile that we need to examine. Moreover, it is, in fact, not necessary to consider *all* adds, drops or switches, but only the *maximally* profitable ones.

Corollary 1. *A profile σ of a given CAG is a pure strategy Nash equilibrium if and only if there are no maximally profitable elementary changes available.*

We describe a strategy profile σ as *A-stable* (*D-stable*, *S-stable*) if it admits no maximally profitable adds (drops, switches); likewise for *AS-stable*, *DS-stable* and so on. Thus, another way of stating Corollary 1 is that a profile is in equilibrium if and only if it is ADS-stable.

3.3 Pure Strategy Nash Equilibrium

The SPMP has been used as a basis for the proof of existence of a pure strategy Nash equilibrium in the aforementioned special cases of games with congestion-averse utilities. However, as we shall see, the previous techniques have been heavily built on additional properties of the particular models that do not hold for more general CAGs.

Specifically, for games with player-specific payoff functions [7] the proof is based on showing the existence of a best-reply improvement path that connects an arbitrary initial point to a Nash equilibrium. Now, any such path consists of profitable switches only (as in these games only singleton strategies are allowed), which make it impossible to use analogous dynamics in general CAGs where players have strategies of different sizes available to them, and thus a player might need to use adds or drops to modify one strategy to another, rather than just switches. In addition, an upper bound on the length of the shortest path as above provided in [7] is not polynomial. Ackermann et al [1] extended this study to deal with player-specific *matroid congestion games*, in which the strategy space of a player consists of the bases of a matroid on the set of resources, and developed a polynomial time algorithm for computing pure equilibria in such games. However, similarly to the singleton case, this dynamics involves switch-type deviations only, as all player strategies are of the same size.

---

[6] For the class of congestion games with player-specific payoff functions [7] and, in fact, for any model with singleton strategies, the property holds trivially, as the set of all possible deviations is restricted to switches only.

The polynomial time procedure developed in [15] for taxed congestion games with failures, is based on the specific property that an add applied to any DS-stable strategy profile either preserves its DS-stability, or requires only one uniquely defined drop for stabilisation. This property, however, does not hold even for other models with faulty/random-order services, where an add may cause a long chain of consecutive drops. Therefore, in games with load-dependent failures [13] and random order congestion games [14], algorithms were developed to first find an initial DS-stable strategy profile whose stability is not affected by adds, termed "post-addition DS-stable profile". In general CAGs however, such a profile does not necessarily exist. In addition, the above algorithms only deal with identical resources and specific utility functions. This implies the need to develop a new, universal technique for computation of equilibria in games with congestion-averse utilities.

We now proceed to investigate the properties of particular sequences of elementary changes, which we call "drop and swap ladders", when applied to partially stable strategy profiles. As we shall see, these ladders play a central role in our general method of constructing pure strategy Nash equilibria.

Drop and Swap Ladders. Suppose that a strategy profile σ is AS-stable, but does have a sequence of profitable deviations, consisting of a drop followed by $m \geq 1$ switches. We define a *drop ladder* to be a sequence as above, all of whose elementary changes are maximally profitable to each of the deviators.

Definition 3. *A* ***drop ladder*** *is a sequence*

$$D_{i_0}(r_0) + S_{i_1}(r'_1 \leftarrow r_1) + \cdots + S_{i_m}(r'_m \leftarrow r_m),$$

consisting of a maximally profitable drop followed by a sequence of $m \geq 0$ maximally profitable switches. The ***length*** *of the ladder is determined by the number of switches, m, and its* ***tail*** *is the last switched-out resource involved, r_m.*

When applied on partially stable profiles, drop ladders have a particular structure and some interesting properties that we summarise in the following lemma.

Lemma 1. *Given a CAG, let σ be an AS-stable strategy profile that possesses a drop ladder of length $m \geq 0$, and let σ^k denote the result of applying the drop and the first k switches to σ. Suppose further that σ^k is A-stable for $0 \leq k < m$. Then,*

- *switches "chain" with one another and with the initial drop: for all $1 \leq k \leq m$ we have $r'_k = r_{k-1}$;*
- *if there is a profitable add $A_i(r_+)$ to the profile σ^m, then $r_+ = r_m$.*

Following the observation made in Lemma 1, we now define an additional class of strategy profile modifications, termed *swap ladders*, as follows.

Definition 4. *A* ***swap ladder*** *is a drop-ladder followed by a maximally profitable add at the end:*

$$D_{i_0}(r_0) + S_{i_1}(r_0 \leftarrow r_1) + \cdots + S_{i_m}(r_{m-1} \leftarrow r_m) + A_{i_{m+1}}(r_m). \quad (1)$$

The number of switches, m, is the ***length*** *of the ladder. The swap ladder is described as* ***minimal*** *if all intermediate strategy profiles before the last add were A-stable (i.e., if the add is performed at the first opportunity).*

By Lemma 1, a profitable add can be made only to the tail of a minimal drop ladder, and the result of the corresponding swap ladder possesses the same congestion vector as the original profile; this gives us a reason to hope that minimal swap ladders preserve AS-stability. To build the intuition for the proof, we first make a couple of observations.

Consider a swap ladder as in (1); let σ^k be the result of applying to σ the drop and the first k switches, and let σ^{m+1} be the final profile after the add. Notice that for any player i_k, $1 \leq k \leq m+1$, who performs the k^{th} move after the initial drop, the congestion on his selected resources immediately before the move, i.e. at σ^{k-1}, is the same as at σ: indeed, the only resource at which congestion is any different from that at σ is r_{k-1}, which at that point is not selected by i_k or he would be unable to switch to or add it. Likewise, after the k^{th} move, at σ^k, player i_k still does not use any resource whose congestion is lower than at σ (if there is any such resource it is the result of i_k switching-out at r_{k+1}, which is clearly no longer part of his profile).

This is the key observation: that within a swap ladder, a player making an elementary change experiences the same congestion immediately before and after the move that he did before the start of the swap ladder. Swap ladders do not change congestion, so in fact this congestion is the same throughout any sequence of swap ladders. More precisely, although congestion does of course change as other players move, the congestion experienced by a given player before and after any move that this particular player makes is the same as in the initial profile, so from his decision-making point of view there is a fixed ranking on resources throughout the sequence of swap ladders. That is, for any particular congestion vector, each player has a ranking on resources determined by the utility of holding that single resource: we say that for player i, $r_1 \leq r_2$, if $U_i(\{r_1\}) \leq U_i(\{r_2\})$. The IIA property then implies that this preference is independent of what other resources the player may have (so long as congestion on r_1 and r_2 does not change): for any $x_i \in \Sigma_i$ such that $r_1, r_2 \notin x_i$, $U_i(x_i \cup \{r_1\}) \leq U_i(x_i \cup \{r_2\})$.

We are now in a position to present the key lemma which is central to our existence proof. It characterises a possible sequence of elementary changes that a given player can make in consecutive minimal swap ladders.

Lemma 2. *Consider the sequence of adds, drops and switches that a single player makes in a sequence of minimal swap ladders. Then,*

– *the resources dropped or switched-out are always the lowest-ranked among the player's selected resources right before the corresponding move,*

– *they form an increasing sequence with respect to the total rank, and*
– *once dropped or switched-out, they are not subsequently added back or switched-in.*

Lemma 2 easily supplies us with a linear bound on the number of changes that a sequence of minimal swap ladders can contain:

Corollary 2. *There can be no more than $2NR$ elementary changes in total in any sequence of minimal swap ladders.*

Proof. Consider a single player's contribution to the sequence of swap ladders. From Lemma 2, once it has been dropped or switched-out, each resource cannot subsequently be added or switched-in; each resource can therefore only be dropped or switched-out once and added or switched-in once. It follows that the total number of elementary changes for a given player is at most $2R$. The result then holds simply by multiplying by the number of players.

Finally, Lemma 2 implies the AS-stability of post-swap-ladder profiles.

Proposition 1. *If a swap ladder is applied to an AS-stable profile σ, then the resulting profile is also AS-stable.*

Based on Proposition 1 and Corollary 2 we then conclude the existence of pure strategy Nash equilibria in CAGs. The following theorem is one of our main contributions.

Theorem 2. *Every CAG possesses a pure strategy Nash equilibrium.*

3.4 Necessity of Congestion-Averseness Conditions

As the congestion-averse conditions on utility functions have been shown to be sufficient to prove the existence of SPMP and pure strategy Nash equilibria, an interesting question that now arises is that of necessity of the above conditions. Below we show that each of the three congestion-averse conditions is necessary, in general, for the existence of SPMP.

Theorem 3. *In a CAG setting, if any one of the congestion-averse conditions on utility functions is violated then the SPMP is not guaranteed to exist.*

4 Computation of Equilibria

We finally make practical use of our theoretical results. The proof of Theorem 2 suggests a constructive algorithm for finding equilibria; we can, starting from any AS-stable profile, look for maximal drop ladders, and convert them into swap ladders whenever the result is not A-stable. Obviously this process must terminate since either the total congestion strictly decreases, or we have a swap ladder, of which—courtesy of Corollary 2—there can only be a limited number consecutively. This algorithm is presented in Algorithm 1.

The analysis of the worst-case asymptotic complexity of Algorithm 1 results in the following proposition.

Algorithm 1. An algorithm for constructing an equilibrium strategy profile, which searches for drop-ladders and swap-ladders in the same loop.

```
INPUT: A CAG game, G
OUTPUT: A pure-strategy equilibrium σ for G
σ ← (R_1, ..., R_N)
while σ is not ASD-stable do
   choose a maximally profitable drop D
   σ ← σ + D
   while σ is not AS-stable do
      find a maximally profitable switch S
      σ ← σ + S
      if σ has a maximally profitable add A then
         σ ← σ + A
      end if
   end while
end while
```

Proposition 2. *Algorithm 1 requires $O(N^2R^2)$ of elementary changes, and has asymptotic complexity $O\left(N^2R^2g(N,R)\right)$, where $g(N,R)$ is the complexity of a player's utility evaluation.*

5 Conclusions

In this paper we have proved the existence of pure strategy Nash equilibria for a large class of games—Games with Congestion-Averse Utilities—loosely modelled on traditional congestion games. We have also provided an algorithm that constructs an equilibrium explicitly. This work was motivated by a desire to address a broader class of resource contention scenarios than those previously modelled, and we have indeed done so; but more remains to be done. In particular, the question of necessity of the congestion-averseness conditions for existence of pure equilibria remains open. This implies a possibility of extending our results to models without the SPMP, which is a great challenge as our current techniques build heavily on this property.

References

1. Ackermann, H., Röglin, H., Vöcking, B.: Pure Nash equilibria in player-specific and weighted congestion games. In: Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C. (eds.) WINE 2006. LNCS, vol. 4286, pp. 50–61. Springer, Heidelberg (2006)
2. Aland, S., Dumrauf, D., Gairing, M., Monien, B., Schoppmann, F.: Exact price of anarchy for polynomial congestion games. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 218–229. Springer, Heidelberg (2006)
3. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: STOC-2005, pp. 67–73 (2005)

4. Daskalakis, C., Papadimitriou, C.: Computing equilibria in anonymous games. In: FOCS-2007, pp. 83–93 (2007)
5. Fabrikant, A., Papadimitriou, C., Talwar, K.: The complexity of pure Nash equilibria. In: STOC-2004, pp. 604–612 (2004)
6. Leyton-Brown, K., Tennenholtz, M.: Local-effect games. In: IJCAI-2003 (2003)
7. Milchtaich, I.: Congestion games with player-specific payoff functions. Games and Economic Behavior 13, 111–124 (1996)
8. Milchtaich, I.: Congestion models of competition. American Naturalist 147(5), 760–783 (1996)
9. Monderer, D.: Solution-based congestion games. Advances in Math. Economics 8, 397–407 (2006)
10. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14, 124–143 (1996)
11. Penn, M., Polukarov, M., Tennenholtz, M.: Congestion games with failures. In: Proceedings of the 6th ACM Conference on Electronic Commerce (EC-2005), pp. 259–268 (2005)
12. Penn, M., Polukarov, M., Tennenholtz, M.: Asynchronous congestion games. In: AAMAS-2008, May 2008, pp. 1605–1608 (2008)
13. Penn, M., Polukarov, M., Tennenholtz, M.: Congestion games with load-dependent failures: Identical resources. Games and Economic Behavior (to appear)
14. Penn, M., Polukarov, M., Tennenholtz, M.: Random order congestion games. Mathematics of Operations Research (to appear)
15. Penn, M., Polukarov, M., Tennenholtz, M.: Taxed congestion games with failures. Annals of Mathematics and Artificial Intelligence (to appear)
16. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory 2, 65–67 (1973)
17. Roughgarden, T., Tardos, E.: How bad is selfish routing. Journal of the ACM 49(2), 236–259 (2002)

A New Derandomization of Auctions

Oren Ben-Zwi, Ilan Newman, and Guy Wolfovitz

Department of Computer Science, University of Haifa,
Mount Carmel, Haifa 31905, Israel
{nbenzv03,ilan,gwolfovi}@cs.haifa.ac.il

Abstract. Let A be a randomized, unlimited supply, unit demand, single-item auction, which given a bid-vector $b \in [h]^n$, has expected profit $\mathbb{E}[P(b)]$. Aggarwal et al. showed that given A, there exists a deterministic auction which given a bid-vector b, guarantees a profit of $\mathbb{E}[P(b)]/4 - O(h)$. In this paper we show that given A, there exists a deterministic auction which given a bid-vector b of length n, guarantees a profit of $\mathbb{E}[P(b)] - O(h\sqrt{n \ln hn})$. As is the case with the construction of Aggarwal et al., our construction is not polynomial time computable.

1 Introduction

For our good fortune, we were hired to design a mechanism for handling the upcoming 'world cup' TV broadcasts. We are given a two sided communication with the (numerous) potential costumers, the marginal cost for adding one viewer is negligible, and our primary goal is to maximize our revenue. The classical approach for maximizing the revenue on scenarios like this is to set up a fixed price, and charge it from any viewer. However, the price can be fixed too high, causing a smaller number of viewers, or too low, causing a low price collecting from each viewer. Either way, the overall revenue might be too low.

This motivates the study of an *unlimited supply, unit demand, single item* auction. These auctions can guarantee a revenue which is a constant approximation to the *best single price revenue* (which is not necessarily known). In this paper we study the derandomization of such *truthful* auctions. Goldberg et al. [5] introduced randomized auctions that achieve on expectation a constant fraction approximation of the optimal single price revenue.[1] They named these auctions *competitive* after the notion of *competitive analysis* of *online algorithms*. They also proved that randomization is essential assuming the auction is symmetric (that is, assuming the outcome of the auction does not depend on the order of the input bids). Aggarwal et al. [1] later showed how to construct from any randomized auction a deterministic, asymmetric auction with approximately the same revenue. More accurately, given a randomized auction A which accepts bid-vectors in $[1, h]^n$, they constructed a deterministic, asymmetric auction A_D satisfying $P_{A_D}(b) \geq \mathbb{E}[P_A(b)]/4 - O(h)$ for every $b \in [1, h]^n$; here $P_{A_D}(b)$ is the

[1] Actually, they looked on optimal single price where there are at least two buyers, see [5] for farther details.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 233–237, 2009.

profit of A_D given a bid-vector b and $\mathbb{E}[P_A(b)]$ is the expected profit of A given a bid-vector b. The same result also holds in the more restrictive case where A accepts bid-vectors in $[h]^n$. In addition, Aggarwal et al. showed that if the bid-vectors are restricted to be vectors of powers of 2 then the multiplicative factor of 4 above can be improved to 2.

In this paper we show that in the case where the bid-vectors come from $[h]^n$, one can improve the construction of A_D above so as to guarantee a better lower bound for $P_{A_D}(b)$, for the cases where $P_{A_D}(b) = \omega(h\sqrt{n \ln hn})$. Formally we prove the following.

Theorem 1. *Let A be a randomized auction which accepts bid-vectors in $[h]^n$. Assume that A has expected profit $\mathbb{E}[P_A(b)]$ for every bid-vector $b \in [h]^n$. Then there exists a deterministic auction A_D that guarantees a profit of $P_{A_D}(b) \geq \mathbb{E}[P_A(b)] - O(h\sqrt{n \ln hn})$ for every bid-vector $b \in [h]^n$.*

The proof of Theorem 1 can be outlined roughly as follows. Given a randomized auction A, we first define a distribution over a set of deterministic auctions. We then show that if we choose a deterministic auction A_D from that distribution at random, then $\mathbb{E}[P_{A_D}(b)] = \mathbb{E}[P_A(b)]$ for every bid-vector b (where the expectancy on the left-hand side is w.r.t. the choice of A_D and the expectancy on the right-hand side is w.r.t. the coin tosses of A). In addition to that, our distribution has the property that the event Bad_b, that $P_{A_D}(b) < \mathbb{E}[P_{A_D}(b)] - t$, depends on a relatively few number of other events $Bad_{b'}$. Moreover, for every b, we have that the probability of Bad_b is sufficiently small. We then apply the Lovász Local Lemma to show that there exists a choice for A_D for which none of the events Bad_b occur. For our choice of t, this will give the theorem.

We stress the fact that the result of Aggarwal et al. [1] is more general in the sense that it deals with bid-vectors in $[1, h]^n$, while Theorem 1 only deals with discrete bid-vectors. Still, discrete bid-vectors make more sense in real life auctions, where for example, bids are being made in Dollars and Cents. We also note that the construction used in the proof of Theorem 1 is not polynomial time computable and that this is also the case in the construction of Aggarwal et al. [1].

2 Preliminaries

Definition 1 (Unlimited Supply, Unit Demand, Single Item Auction). *An* unlimited supply, unit demand, single item auction *is a mechanism in which there is one* item *of unlimited supply to sell by an* auctioneer *to n* bidders*. The bidders place* bids *for the item according to their* valuation *of the item. The auctioneer then sets prices for every bidder. If the price for a bidder is lower than or equal to its bid, then the bidder is considered as a* winner *and gets to buy the item for its price. A bidder with price higher than its bid does not pay nor gets the item. The auctioneer's* profit *is the sum of the winners prices.*

For a natural number k, let $[k]$ denote the set $\{1, 2, ..., k\}$. A bid-vector $b \in [h]^n$ is a vector of n bids in $[h]$. For $b \in [h]^n$ we denote by b_{-i} the vector which is the result of replacing the ith bid in b with a question mark; that is, b_{-i} is the vector $(b_1, b_2, \ldots, b_{i-1}, ?, b_{i+1}, \ldots b_n)$.

A *truthful auction* is an auction in which every bidder bids its true valuation for the item. It is well known that truthfulness can be achieved through *bid-independent* auctions (see for example [4]). A *bid-independent* auction is an auction for which the auctioneer computes the price for bidder i using only the vector b_{-i} (that is, without the ith bid).

2.1 A Structural Lemma

Let A be a randomized truthful auction that accepts bid-vectors from $[h]^n$. We can view A's execution in the following manner. The auction maintains a set of nm functions $\{g_{i,j}\colon i \in [n], j \in [m]\}$, where $g_{i,j}$ is a function from vectors in $([h] \cup \{?\})^n$ with exactly one question mark to $[h]$. On a bid-vector $b \in [h]^n$, the auction tosses some coins, and chooses accordingly an integer $j \in [m]$. We let p_j be the probability that $j \in [m]$ was chosen. The auction then offers bidder i the price $g_{i,j}(b_{-i})$. Let $\text{accept}_{i,j}(b)$ be 1 if $g_{i,j}(b_{-i}) \leq b_i$ and 0 otherwise. The expected profit of the auction on input b is then:

$$\mathbb{E}[P_A(b)] = \sum_j p_j \sum_i \text{accept}_{i,j}(b) \cdot g_{i,j}(b_{-i}).$$

One can define the following randomized auction A', which is equivalent to the above randomized auction A with respect to expected profits. First, A' maintains the exact same list of functions as A. On a bid-vector $b \in [h]^n$, the auction performs the following independently for every $i \in [n]$: it tosses the same coins that A does, chooses accordingly an integer $j \in [m]$ and then offers the ith bidder price $g_{i,j}(b_{-i})$. The expected profit of A' on input b is given by:

$$\mathbb{E}[P_{A'}(b)] = \sum_i \sum_j p_j \cdot \text{accept}_{i,j}(b) \cdot g_{i,j}(b_{-i}).$$

We call A' the *bidder-self-randomness-dual* of A. The following clearly follows from the discussion above.

Lemma 1. *Let A be a randomized auction and A' be its bidder-self-randomness-dual auction. Then A and A' have the same expected profit on every bid-vector.*

2.2 Probabilistic Tools

The proof of Theorem 1 makes use of the Lovász Local Lemma [3]. We need the following version of the lemma [2].

Lemma 2 (The Local Lemma; Symmetric Case). *Let Bad_i, $1 \leq i \leq N$, be events in an arbitrary probability space. Suppose that each event Bad_i is mutually independent of a set of all the other events Bad_j but at most d, and that $\Pr[Bad_i] \leq p$ for all $1 \leq i \leq N$. If $ep(d+1) \leq 1$, where e is the base of the natural logarithm, then $\Pr[\bigwedge_{i=1}^{N} \neg Bad_i] > 0$.*

Let X be the average of n independent random variables X_i, where $X_i \in [a_i, b_i]$ for all i. We will need the following inequality [6]:

Lemma 3 (Hoeffding). $\Pr[X < \mathbb{E}[X] - t] \le 2\exp\left(\frac{-2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$

3 Proof of Theorem 1

Let A be a randomized auction which accepts bid-vectors in $[h]^n$. Let $\{g_{i,j}\colon i \in [n], j \in [m]\}$ be the set of functions that A maintains. We construct a deterministic auction A_D. To do that, we first define a certain tripartite graph $G = (\mathit{Lft}, \mathit{Cntr}, \mathit{Rght}, E)$. Using this tripartite graph, we define a distribution over deterministic auctions. We will then obtain a deterministic auction A_D by choosing an auction at random according to that distribution. The proof of Theorem 1 will follow by showing that A_D satisfies the conclusion in the theorem with positive probability.

We first describe the tripartite graph G. We let Lft be the set of all nh^{n-1} vectors b_{-i}, where $b \in [h]^n$ is a bid-vector and $i \in [n]$. We let Cntr be the set of all pairs $\{(b_{-i}, g_{i,j}(b_{-i})) : i \in [n], j \in [m]\}$. We let Rght be the set of all possible h^n bid-vectors in $[h]^n$. The edges E are defined as follows. A vertex $b_{-i} \in \mathit{Lft}$ is connected to all the vertices in the set $\{(b_{-i}, g_{i,j}(b_{-i})) \in \mathit{Cntr}\colon j \in [m]\}$. A vertex $(b_{-i}, g_{i,j}(b_{-i})) \in \mathit{Cntr}$ is connected to all bid-vectors $r \in \mathit{Rght}$ for which it holds that $b_{-i} = r_{-i}$ and $\mathrm{accept}_{i,j}(r) = 1$.

Observe that every subgraph G' of G in which every $b_{-i} \in \mathit{Lft}$ has exactly one adjacent edge induces naturally a deterministic auction A_D. To see that this is indeed the case, consider such a subgraph G' of G. The deterministic auction A_D behaves as follows: on a bid-vector $b \in [h]^n$, the price offered to the ith bidder is $g_{i,j}(b_{-i})$ if and only if $\{b_{-i}, (b_{-i}, g_{i,j}(b_{-i}))\}$ is an edge in G'.

Let G' be a subgraph of G chosen in the following way. Independently, for every $b_{-i} \in \mathit{Lft}$, choose a random edge $\{b_{-i}, (b_{-i}, g_{i,j}(b_{-i}))\}$ according to the distribution $\{p_j\}_{j=1}^{m}$. Let A_D be the deterministic auction that is naturally induced by G'. Note that for every bid-vector $b \in [h]^n$,

$$\mathbb{E}[P_{A_D}(b)] = \sum_i \sum_j p_j \cdot \mathrm{accept}_{i,j}(b) \cdot g_{i,j}(b_{-i}),$$

which by Lemma 1, is equal to $\mathbb{E}[P_A(b)]$.

Let Bad_b be the event that $P_{A_D}(b) < \mathbb{E}[P_{A_D}(b)] - t$, where we define $t := 10h\sqrt{n \ln hn}$. We need the following two claims.

Claim. For all $b \in [h]^n$, $\Pr[\mathit{Bad}_b] < 1/(10hn)$.

Proof. Fix $b \in [h]^n$ and let X_i be the profit extracted from bidder i, that is, $X_i = \mathrm{accept}_{i,j}(b) \cdot g_{i,j}(b_{-i})$ (recall that j is determined by A_D). Note that $X_i \in [1, h]$ for all i and that the X_i's are independent random variables. Let X be the average of the X_i's. We have

$$\Pr[\mathit{Bad}_b] = \Pr[P_{A_D}(b) < \mathbb{E}[P_{A_D}(b)] - t] = \Pr[X < \mathbb{E}[X] - t/n],$$

which by Lemma 3 is at most $2\exp\left(\frac{-2t^2}{h^2 n}\right)$. The claim now follows since $t = 10h\sqrt{n \ln hn}$. □

Claim. For all $b \in [h]^n$, Bad_b depends on at most hn other events $Bad_{b'}$.

Proof. Fix $b \in [h]^n$. It is enough to show that there are at most hn vertices $b' \in Rght$ with the following property: there is a vertex $b_{-i} \in Lft$ such that there is a path of length 2 from b_{-i} to b and from b_{-i} to b'. Indeed, for the vertex $b \in Rght$, there are at most n vertices $b_{-i} \in Lft$ which are at distance 2 from b. In addition, for every $b_{-i} \in Lft$ there are at most h vertices $b' \in Rght$ which are at distance 2 from b_{-i}. □

Combining the two claims above with the Lovász Local Lemma, we get that with positive probability Bad_b does not occur for all $b \in [h]^n$. Hence, with positive probability, for every bid-vector $b \in [h]^n$, $P_{A_D}(b) \geq \mathbb{E}[P_{A_D}(b)] - t$. This proves the theorem.

References

1. Aggarwal, G., Fiat, A., Goldberg, A.V., Hartline, J.D., Immorlica, N., Sudan, M.: Derandomization of auctions. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), pp. 619–625 (2005)
2. Alon, N., Spencer, J.: The Probabilistic Method, 3rd edn. Wiley, New-York (2008)
3. Erdős, P., Lovász, L.: Problems and results on 3-chromatic hypergraphs and some related questions. In: Hajnal, A., et al. (eds.) Infinite and Finite Sets, pp. 609–628 (1975)
4. Fiat, A., Goldberg, A.V., Hartline, J.D., Karlin, A.: Competitive generalized auctions. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC). ACM Press, New York (2002)
5. Goldberg, A.V., Hartline, J.D., Wright, A.: Competitive auctions and digital goods. In: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 735–744 (2001)
6. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association 58(301), 13–30

The Computational Complexity of Weak Saddles

Felix Brandt, Markus Brill, Felix Fischer, and Jan Hoffmann

Institut für Informatik, Ludwig-Maximilians-Universität München
80538 München, Germany
{brandtf,brill,fischerf,hoffmann}@tcs.ifi.lmu.de

Abstract. We continue the recently initiated study of the computational aspects of weak saddles, an ordinal set-valued solution concept proposed by Shapley. Brandt et al. gave a polynomial-time algorithm for computing weak saddles in a subclass of matrix games, and showed that certain problems associated with weak saddles of bimatrix games are NP-complete. The important question of whether weak saddles can be *found* efficiently was left open. We answer this question in the negative by showing that finding weak saddles of bimatrix games is NP-hard, under polynomial-time Turing reductions. We moreover prove that recognizing weak saddles is coNP-complete, and that deciding whether a given action is contained in some weak saddle is hard for parallel access to NP and thus not even in NP unless the polynomial hierarchy collapses. Our hardness results are finally shown to carry over to a natural weakening of weak saddles.

1 Introduction

Saddle points, i.e., combinations of actions such that no player can gain by deviating, are one of the earliest solutions suggested in game theory (see, e.g., [25]). In two-player zero-sum games (henceforth *matrix games*), every saddle point happens to coincide with an optimal outcome both players can guarantee in the worst case and thus enjoys a very strong normative foundation. Unfortunately, however, not every matrix game possesses a saddle point. In order to remedy this situation, von Neumann [24] considered *mixed*, i.e., randomized, strategies and proved that every matrix game contains a mixed saddle point (or equilibrium) that moreover maintains the appealing normative properties of saddle points. The existence result was later generalized to arbitrary general-sum games by Nash [17], at the expense of its normative foundation. Since then, Nash equilibrium has commonly been criticized for its need for randomization, which may be deemed unsuitable, impractical, or even infeasible (see, e.g., [14, 15, 5]).

In two papers from 1953, Lloyd Shapley showed that existence of saddle points (and even uniqueness in the case of matrix games) can also be guaranteed by moving to *minimal sets* of actions rather than randomizations over them [21, 22].[1] Shapley defines a *generalized saddle point (GSP)* to be a tuple of subsets of actions of each player, such that every action not contained in the GSP is

[1] The main results of the 1953 reports later reappeared in revised form [23].

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 238–249, 2009.

dominated by some action in the GSP, given that the remaining players choose actions from the GSP. A *saddle* is an inclusion-minimal GSP, i.e., a GSP that contains no other GSP. Depending on the underlying notion of dominance, one can define strict, weak, and very weak saddles. Shapley [23] showed that every matrix game admits a *unique* strict saddle. Duggan and Le Breton [10] proved that the same is true for the weak saddle in a certain subclass of symmetric matrix games that we refer to as *confrontation games*. While Shapley was the first to conceive weak GSPs, he was not the only one. Apparently unaware of Shapley's work, Samuelson [20] uses the very related concept of a *consistent pair* to point out epistemic inconsistencies in the concept of iterated weak dominance. Also, *weakly admissible sets* as defined by McKelvey and Ordeshook [15] in the context of spatial voting games are identical to weak GSPs. Other common *set-valued* concepts in game theory include *rationalizability* [3, 19] and *CURB sets* [1] (see also Myerson's textbook ([16], pp. 88-91) for a general discussion of set-valued solution concepts).

In this paper we continue the recently initiated study of the computational aspects of Shapley's saddles. Brandt et al. [5] gave polynomial-time algorithms for computing strict saddles in general games and weak saddles in confrontation games. Although it was shown that certain problems associated with weak saddles in bimatrix games are NP-complete, the question of whether weak saddles can be found efficiently was left open. We answer this question in the negative by showing that finding weak saddles is NP-hard. Moreover, we prove that recognizing weak saddles is coNP-complete, and that deciding whether an action is contained in a weak saddle of a bimatrix game is complete for parallel access to NP and thus not even in NP unless the polynomial hierarchy collapses. We finally demonstrate that our hardness results carry over to very weak saddles.

2 Related Work

In recent years, the computational complexity of game-theoretic solution concepts has come under increasing scrutiny. One of the most prominent results in this stream of research is that the problem of finding Nash equilibria in bimatrix games is PPAD-complete [7, 9], and thus unlikely to admit a polynomial-time algorithm. PPAD is a subclass of FNP, and it is obvious that Nash equilibria can be recognized in polynomial time. Interestingly, our results imply that this is not the case for weak saddles unless P=NP.

Weak saddles rely on the elementary concept of weak dominance, whose computational aspects have been studied extensively in the form of *iterated* weak dominance [12, 8, 6]. In contrast to iterated dominance, saddles are based on a notion of stability reminiscent of Nash equilibrium and its various refinements. Weak saddles are also related to minimal covering sets, a concept that has been proposed independently in social choice theory [11, 10] and whose computational complexity has recently been analyzed [4, 2].

Brandt et al. [5] constructed a class of games that established a strong relationship between weak saddles and inclusion-maximal cliques in undirected graphs. Based on this construction and a reduction from the NP-complete problem CLIQUE, they showed that deciding whether there exists a weak saddle with a *certain number of actions* is NP-hard. This construction, however, did not permit any statements about the more important problems of *finding* a weak saddle, *recognizing* a weak saddle, or *deciding whether a certain action is contained* in some weak saddle.

3 Preliminaries

An accepted way to model situations of strategic interactions is by means of a *normal-form game* (see, e.g., [14]).

Definition 1 (Normal-Form Game). *A (finite)* game in normal-form *is a tuple $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ where $N = \{1, 2, \ldots, n\}$ is a set of* players *and for each player $i \in N$, A_i is a nonempty finite set of* actions *available to player i, and $p_i : (\prod_{i\in N} A_i) \to \mathbb{R}$ is a function mapping each action profile (i.e., combination of actions) to a real-valued* payoff *for player i.*

A *subgame* of a (normal-form) game $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ is a game $\Gamma' = (N, (A'_i)_{i\in N}, (p'_i)_{i\in N})$ where, for each $i \in N$, A'_i is a nonempty subset of A_i and $p'_i(a') = p_i(a')$ for all $a' \in A'_1 \times \ldots \times A'_n$. Γ is then called a *supergame* of Γ'.

In order to formally define Shapley's weak saddles, we need some additional notation. Let $A_N = (A_1, \ldots, A_n)$. For a tuple $S = (S_1, \ldots, S_n)$, write $S \subseteq A_N$ and say that S is a subset of A_N if $\emptyset \neq S_i \subseteq A_i$ for all $i \in N$. Further let $S_{-i} = (S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_n)$. For a player $i \in N$ and two actions $a_i, b_i \in A_i$ say that a_i *weakly dominates* b_i with respect to S_{-i}, denoted $a_i >_{S_{-i}} b_i$, if $p_i(a_i, s_{-i}) \geq p_i(b_i, s_{-i})$ for all $s_{-i} \in S_{-i}$, with at least one strict inequality.

Definition 2 (Weak Saddle). *Let $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$ be a game and $S = (S_1, \ldots, S_n) \subseteq A_N$. Then, S is a* weak generalized saddle point (WGSP) *of Γ if for each player $i \in N$ the following holds:*

$$\text{For every } a_i \in A_i \setminus S_i \text{ there exists } s_i \in S_i \text{ such that } s_i >_{S_{-i}} a_i. \tag{1}$$

A weak saddle *is a WGSP that contains no other WGSP.*

An example game with two weak saddles is given in Figure 1. The interpretation of this definition is the following: Every player i has a distinguished set S_i of actions such that for every action a_i that is not in the set S_i, there is some action in S_i that weakly dominates a_i, provided that the other players play only actions from their distinguished sets. Condition (1) will be called *external stability* in the following. A WGSP thus is a tuple S that is externally stable for each player. Observe that the tuple A_N of all actions is always a WGSP, thereby guaranteeing existence of a weak saddle in every game. As the game in Figure 1 illustrates, weak saddles do not have to be unique. It is also not very hard to see that weak saddles are invariant under order-preserving transformations of the payoff functions and that every weak saddle contains a (mixed) Nash equilibrium.

	b_1	b_2	b_3
a_1	0 2	1 1	0 1
a_2	1 1	1 1	1 2

Fig. 1. Example game with two weak saddles: $(\{a_1\}, \{b_1, b_2\})$ and $(\{a_1, a_2\}, \{b_2\})$

In the remainder of the paper we will concentrate on two-player games.[2] For such games, we can simplify notation and write $\Gamma = (A, B, p)$, where A is the set of actions of player 1, B is the set of actions of player 2, and $p : A \times B \to \mathbb{R} \times \mathbb{R}$ is the payoff function on the understanding that $p(a, b) = (p_1(a, b), p_2(a, b))$ for all $(a, b) \in A \times B$. A two-player game is often called a *bimatrix* game, as it can conveniently be represented as a $|A| \times |B|$ bimatrix M, i.e., a matrix with rows indexed by A, columns indexed by B and $M(a, b) = p(a, b)$ for every action profile $(a, b) \in A \times B$. We will commonly refer to actions of players 1 and 2 by the rows and columns of this matrix, respectively. When representing a bimatrix game graphically, we follow the convention to write player 1's payoffs in the lower left corner and player 2's payoff in the upper right corner of the corresponding matrix cell (see Figure 1 for an example).

For an action a and a weak saddle $S = (S_1, S_2)$, we will sometimes slightly abuse notation and write $a \in S$ if $a \in (S_1 \cup S_2)$. In such cases, whether a is a row or a column should be either clear from the context or irrelevant for the argumentation. This partial identification of S and $S_1 \cup S_2$ is also reflected in referring to S as a "set" rather than a "pair" or "tuple." When reasoning about the structure of the saddles of game, the following notation will be useful. For two actions $x, y \in A \cup B$, we write $x \rightsquigarrow y$ if every weak saddle containing x also contains y. Observe that $\rightsquigarrow$ as a relation on $(A \cup B) \times (A \cup B)$ is transitive. We now identify two sufficient conditions for $x \rightsquigarrow y$ to hold.

Fact 1. *Let $\Gamma = (A, B, p)$ be a two-player-game, $b \in B$ an action of player 2, and $a \in A$ an action of player 1. Then $b \rightsquigarrow a$ if one of the following two conditions holds.*[3]

(i) a is the unique action maximizing $p_1(\cdot, b)$, i.e., $\{a\} = \arg\max_{a' \in A} p_1(a', b)$.
(ii) a maximizes $p_1(\cdot, b)$, and all actions maximizing $p_1(\cdot, b)$ yield identical payoffs for all *opponent actions, i.e., $a \in \arg\max_{a' \in A} p_1(a', b)$ and $p_1(a_1, b') = p_1(a_2, b')$ for all $a_1, a_2 \in \arg\max_{a' \in A} p_1(a', b)$ and all $b' \in B$.*

Part *(i)* of the statement above can be generalized in the following way. An action a is in the weak saddle if it is a unique best response to a subset of saddle actions: if $\{b_1, \ldots, b_t\} \subset S$ and there is no $a' \in A \setminus \{a\}$ with $p_1(a', b_i) \geq p_1(a, b_i)$ for all $i \in [t]$, then $a \in S$.[4] In this case, we write $\{b_1, \ldots, b_t\} \rightsquigarrow a$. Moreover, for two

[2] Naturally, all hardness results carry over to the general n-player case by adding an arbitrary number of "dummy" players that always receive the same payoff.

[3] The statement remains true if the roles of the two players are reversed.

[4] For $n \in \mathbb{N}$, we write $[n] = \{1, 2, \ldots, n\}$.

sets of actions X and Y, we write $X \rightsquigarrow Y$ if $X \rightsquigarrow y$ for all $y \in Y$. For example, in the game in Figure 1, $b_1 \rightsquigarrow a_1 \rightsquigarrow b_2$, $\{b_2, b_3\} \rightsquigarrow a_2$ and $\{b_1, b_3\} \rightsquigarrow \{a_1, a_2\}$.

We assume throughout the paper that games are given explicitly, i.e., as tables containing the payoffs for every possible action profile. We will be interested in the following computational problems for a given game Γ:

- FindWeakSaddle: Find a weak saddle of Γ.
- IsWeakSaddle: Is a given collection $(S_1, \ldots, S_n)$ of subsets of actions for each player a weak saddle of Γ?
- UniqueWeakSaddle: Does Γ contain exactly one weak saddle?
- InWeakSaddle: Is a given action a contained in a weak saddle of Γ?
- InAllWeakSaddles: Is a given action a contained in *every* weak saddle of Γ?
- NontrivialWeakSaddle: Does Γ contain a weak saddle that does *not* consist of all actions?

We assume the reader to be familiar with the basic notions of complexity theory, such as polynomial-time many-one reductions and Turing reductions, and the related notions of hardness and completeness, and with standard complexity classes such as P, NP, and coNP (see, e.g., [18]). We will further use the complexity classes Σ_2^p and Θ_2^p. $\Sigma_2^p = \text{NP}^{\text{NP}}$ forms part of the second level of the polynomial hierarchy and consists of all problem that can be solved on a non-deterministic Turing machine with access to an NP oracle. $\Theta_2^p = \text{P}_{||}^{\text{NP}}$ consists of all problems that can be solved on a deterministic Turing machine with parallel (non-adaptive) access to an NP oracle.

4 Hardness Results for Weak Saddles

We will now derive various hardness results for weak saddles. We begin by presenting a general construction that transforms a Boolean formula φ into a bimatrix game Γ_φ, such that the existence of certain weak saddles in Γ_φ depends on the satisfiability of φ. This construction will be instrumental for each of the hardness proofs. The proofs themselves are often omitted due to space constraints and can be found in the full version of this paper.

4.1 A General Construction

Let $\varphi = C_1 \wedge \ldots \wedge C_m$ be a Boolean formula in conjunctive normal form (CNF) over a finite set $V = \{v_1, \ldots, v_n\}$ of variables. Denote by $L = \bigcup_{v \in V} \{\{v, \overline{v}\} : v \in V\}$ the set of all *literals*, where a literal is either a variable or its negation. Each *clause* C_j is a set of literals. An *assignment* $\alpha : L \to \{0, 1\}$ is a function mapping each literal to either 1 (“true”) or 0 (“false”). Assignment α is *valid* if $\alpha(v) \neq \alpha(\overline{v})$ for all $v \in V$. For a valid assignment α, denote by $L^\alpha = \{\ell \in L : \alpha(\ell) = 1\}$ the set of literals that are set to true under α. We say that α *satisfies* a clause C_j if $C_j \cap L^\alpha \neq \emptyset$. Finally, formula φ is *satisfiable* if there is an assignment that satisfies each of its clauses. We assume without loss of generality that φ

	b^*	v_1	$\overline{v}_1$	v_2	$\overline{v}_2$	$\ldots$	v_n	$\overline{v}_n$
a^*	1 1	0 0	0 0	0 0	0 0	$\ldots$	0 0	0 0
d^*	0 0	1 1	1 1	1 1	1 1	$\ldots$	1 1	1 1
C_1	1 0	0 0	0 1	0 1	0 0	$\ldots$	0 1	0 1
C_2	1 0	0 1	0 0	0 1	0 1	$\ldots$	0 0	0 1
$\vdots$					$\vdots$			
C_m	1 0	0 1	0 1	0 1	0 1	$\ldots$	0 1	0 1

Fig. 2. Subgame of Γ_φ for a formula $\varphi = C_1 \wedge \cdots \wedge C_m$ with $v_1, \overline{v}_2 \in C_1$ and $\overline{v}_1, v_n \in C_2$

does not contain *trivial* clauses, i.e., clauses that contain a literal ℓ as well as its negation $\overline{\ell}$. The game $\Gamma_\varphi = (A, B, p)$ is defined in three steps.

Step 1. Player 1 has actions $\{a^*, d^*\} \cup C$, where $C = \{C_1, \ldots, C_m\}$ is the set of clauses of φ. Player 2 has actions $B = \{b^*\} \cup L$, where L is the set of literals.[5] Payoffs are given by

- $p(a^*, b^*) = (1, 1)$,
- $p(d^*, \ell) = (1, 1)$ for all $\ell \in L$,
- $p(C_j, b^*) = (0, 1)$ for all $j \in [m]$,
- $p(C_j, \ell) = (1, 0)$ if and only if $\ell \notin C_j$,
- $p(a, b) = (0, 0)$ otherwise.

An example of such a game is shown in Figure 2. Observe that (a^*, b^*) is a weak saddle and thus no strict superset can be a weak saddle. Furthermore, row d^* dominates row C_j with respect to a set of columns $\{\ell_1, \ldots, \ell_t\} \subseteq L$ if and only if $\ell_i \in C_j$ for some $i \in [t]$. In particular, $d^* >_{L^\alpha} C_j$ if and only if α satisfies C_j. Another noteworthy property of this game is the fact that no weak saddle contains any of the rows C_j, because $C_j \rightsquigarrow b^* \rightsquigarrow a^*$ for each $j \in [m]$.

The basic idea behind this construction is the following. We want to have an "assignment saddle" $S^\alpha = (S_1, S_2)$ with $d^* \in S_1$ and $S_2 = L^\alpha$ if and only if α satisfies φ. For the direction from left to right, we have to ensure that S^α cannot be a weak saddle if α does not satisfy φ or if α is not a valid assignment. This is achieved by means of additional actions, for which the payoffs are defined in such a way that every "wrong" (i.e., unsatisfying or invalid) assignment yields a set containing both a^* and b^*. Obviously, such a set can never be a weak saddle, because it contains the weak saddle (a^*, b^*) as a proper subset. In fact, (a^*, b^*) will be the unique weak saddle in cases where there is no satisfying assignment.

[5] There shall be no confusion by identifying literals with corresponding actions of player 2, which will henceforth be called "literal actions" (or "literal columns").

Step 2. We augment the action sets of both players. Player 1 has one additional row ℓ' for each literal $\ell \in L$.[6] Player 2 has one additional column y_i for each variable $v_i \in V$. Payoffs for profiles involving new actions are defined as follows:

- $p(a^*, y_i) = (1, 0)$ for all $i \in [n]$,
- $p(\ell', \ell) = (2, 1)$ if $\ell' = \ell$,
- $p(\ell', y_i) = (0, 1)$ if $\ell' \in \{v_i, \overline{v}_i\}$,
- $p(a, b) = (0, 0)$ otherwise.

Observe that, by Fact 1 and the discussion thereafter, $\ell \rightsquigarrow \ell'$, $\{\ell', \overline{\ell'}\} \rightsquigarrow y_i$ and $y_i \rightsquigarrow a^* \rightsquigarrow b^*$. This means that no assignment saddle can contain both ℓ as well as its negation $\overline{\ell}$.

There only remains one subtlety to be dealt with. In the game defined so far, there are weak saddles containing row d^*, whose existence is independent of the satisfiability of φ, namely $(\{d^*, \ell'\}, \{\ell\})$ for each $\ell \in L$. We destroy these saddles using additional rows.

Step 3. We introduce new rows $r_1, \overline{r}_1, \dots, r_n, \overline{r}_n$, one for each literal, with the property that $r_i \rightsquigarrow b^*$, and that r_i or $\overline{r}_i$ can only be weakly dominated (by v_i and $\overline{v}_i$, respectively) if at least one literal column other than v_i or $\overline{v}_i$ is in the saddle. For this, we define

- $p(r_i, b^*) = p(\overline{r}_i, b^*) = (0, 1)$ for all $i \in [n]$,
- $p(r_i, v_i) = r(\overline{r}_i, \overline{v}_i) = (2, 0)$,
- $p(r_i, \ell) = p(\overline{r}_i, \ell) = (-1, 0)$ if $\ell \in \{v_{i+1}, \overline{v}_{i+1}\}$ (where $v_{n+1} = v_1$),
- $p(a, b) = (0, 0)$ otherwise.

The game Γ_φ now has action sets $A = \{a^*, d^*\} \cup C \cup L \cup \{r_1, \dots, \overline{r}_n\}$ for player 1 and $B = \{b^*\} \cup L \cup \{y_1, \dots, y_n\}$ for player 2. The size of Γ_φ thus is clearly polynomial in the size of φ. A complete example of such a game is given in the full version of this paper.

For an assignment α, define the assignment saddle S^α as $S^\alpha = (\{d^*\} \cup L^\alpha, L^\alpha)$. It should be clear from the argumentation above that S^α is a weak saddle of Γ_φ if and only if α satisfies φ.

4.2 Membership Is NP-hard

We show NP-hardness of the membership problem via a reduction from SAT. Given a CNF formula φ, we show that the game Γ_φ defined in Section 4.1 has a weak saddle containing action d^* if and only if φ is satisfiable. In particular, we need to show that there is no saddle containing d^* if φ is unsatisfiable.

Theorem 1. InWeakSaddle *is NP-hard.*

[6] Action ℓ' of player 1 and action ℓ of player 2 refer to the same literal, but we name them differently to avoid confusion.

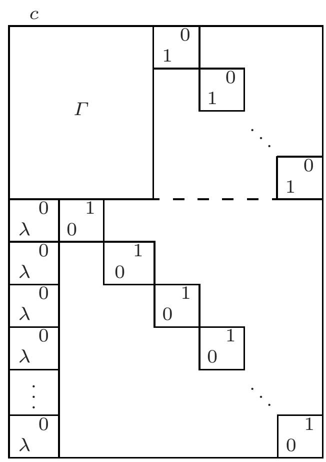


Fig. 3. Construction used in the proof of Lemma 1. Payoffs are $(0, 0)$ unless specified otherwise, λ is chosen to maximize $p_1(\cdot, c)$. Every weak saddle containing column c then equals the set of all actions.

4.3 Membership Is coNP-hard

In order to show that InWeakSaddle is also coNP-hard, we first show the following: given a game and an action c, it is possible to augment the game with additional actions such that every weak saddle of the augmented game that contains c contains all actions of this game.

Lemma 1. *Let $\Gamma = (A, B, p)$ be a two-player game, $c \in A \cup B$ and action of Γ. Then there exists a supergame $\Gamma^c = (A', B', p')$ of Γ with the following properties:*

(i) If S is a weak saddle of Γ^c containing c, then $S = (A', B')$.
(ii) If S is a weak saddle of Γ that does not *contain c, then S is a weak saddle of Γ^c.*
(iii) The size of Γ^c is polynomial in the size of Γ.

The game Γ^c is sketched in Figure 3. Briefly, we introduce new actions $(A' \setminus A)$ and $(B' \setminus B)$ and define the payoffs for profiles involving these actions in such a way that $c \rightsquigarrow (A' \setminus A) \rightsquigarrow (B' \setminus \{c\}) \rightsquigarrow A$. We have the following.

Theorem 2. InWeakSaddle *is coNP-hard.*

Proof. We give a reduction from UNSAT. For a given CNF formula φ, consider the game $\Gamma_\varphi^{b^*}$ obtained by augmenting the game Γ_φ defined in Section 4.1 in such a way that every weak saddle containing action b^* in fact contains all actions. We show that $\Gamma_\varphi^{b^*}$ has a weak saddle containing b^* if and only if φ is unsatisfiable.

For the direction from left to right, assume that there exists a weak saddle $S = (S_1, S_2)$ with $b^* \in S_2$. By Lemma 1, S is trivial, i.e., equals the set of all actions. Furthermore, S must be the unique weak saddle of $\Gamma_\varphi^{b^*}$, because any other weak saddle would violate minimality of S. In particular, S^α cannot be a saddle for any assignment α, which by the discussion in Section 4.1 means that φ is unsatisfiable.

For the direction from right to left, assume that φ is unsatisfiable. It is not very hard to see that every weak saddle $S = (S_1, S_2)$ contains at least one column not corresponding to a literal, i.e., $S_2 \nsubseteq L$ (otherwise, S would be an assignment saddle). However, since $a^* \rightsquigarrow b^*$ and $b \rightsquigarrow a^*$ for every non-literal column $b \in B \setminus L$, we have that $b^* \in S_2$ for every weak saddle S. □

The proof of Theorem 2 implies several other hardness results.

Corollary 1. *The following holds:*

- IsWeakSaddle *is coNP-complete.*
- InAllWeakSaddles *is coNP-complete.*
- UniqueWeakSaddle *is coNP-hard.*

Proof. Let φ be a Boolean formula, which without loss of generality we can assume to have either no satisfying assignment or more than one. Otherwise, we could add a clause $\{w_1, w_2\}$ to φ (where w_1 and w_2 are new variables), thereby multiplying the number of satisfying assignments by three. Recall the definition of the game $\Gamma_\varphi^{b^*}$ used in the proof of Theorem 2. It is easily verified that the following statements are equivalent: formula φ is unsatisfiable, $\Gamma_\varphi^{b^*}$ has a trivial weak saddle, $\Gamma_\varphi^{b^*}$ has a unique weak saddle, b^* is contained in all weak saddles of $\Gamma_\varphi^{b^*}$. This provides a reduction from UNSAT to each of the problems above.

Membership of InAllWeakSaddles in coNP holds because any externally stable set that does not contain the action in question serves as a witness that this actions is *not* contained in every weak saddle. For membership of IsWeakSaddle, consider a tuple S of actions that is *not* a weak saddle. Then either S itself is not externally stable, or a proper subset of S is. For both cases there exists a witness of polynomial size. □

4.4 Finding a Saddle Is NP-hard

A particularly interesting consequence of Theorem 2 concerns the existence of a nontrivial weak saddle. As we will see, hardness of deciding the latter can be used to obtain a result about the complexity of the search problem.

Corollary 2. NontrivialWeakSaddle *is NP-complete.*

Proof. For membership in NP, observe that proving the existence of a nontrivial weak saddle is tantamount to finding a proper subset of the set of all actions that is externally stable. By definition, every such subset is guaranteed to contain a weak saddle. Obviously, external stability can be checked in polynomial time.

Hardness is again straightforward from the proof of Theorem 2, since the game $\Gamma_\varphi^{b^*}$ has a nontrivial weak saddle if and only if formula φ is satisfiable. □

Corollary 3. FINDWEAKSADDLE *is NP-hard under polynomial-time Turing reductions.*

Proof. Suppose there exists an algorithm that computes some weak saddle of a game in time polynomial in the size of the game. Such an algorithm could obviously be used to solve the NP-hard problem NONTRIVIALWEAKSADDLE in polynomial time. Just run the algorithm once. If it returns a nontrivial saddle, the answer is "yes." Otherwise the set of all actions must be the unique weak saddle of the game, and the answer is "no." □

4.5 Membership Is Θ_2^p-hard

Now that we have established that INWEAKSADDLE is both NP-hard and coNP-hard, we will raise the lower bound to Θ_2^p. Wagner provided a sufficient condition for Θ_2^p-hardness that turned out to be very useful (see, e.g., [13]).

Lemma 2 (Wagner [26]). *Let S be an NP-complete problem, and let T be any set. Further let f be a polynomial-time computable function such that the following holds for all $k \geq 1$ and all strings $x_1, x_2, \ldots, x_{2k}$ satisfying $x_{j-1} \in S$ whenever $x_j \in S$ for every j with $1 < j \leq 2k$:*

$$\|\{i : x_i \in S\}\| \text{ is odd} \iff f(x_1, x_2, \ldots, x_{2k}) \in T.$$

Then T is Θ_2^p-hard.

The following statement can be shown by applying Wagner's Lemma to the NP-complete problem $S = \text{SAT}$ and to $T = \text{INWEAKSADDLE}$.

Theorem 3. INWEAKSADDLE *is Θ_2^p-hard.*

We conclude this section by showing that Σ_2^p is an upper bound for the membership problem.

Proposition 1. INWEAKSADDLE *is in Σ_2^p.*

Proof. Let $\Gamma = (N, (A_i)_{i \in N}, (p_i)_{i \in N})$ be a game, $d^* \in \bigcup_i A_i$ a designated action. First observe that we can verify in polynomial time whether a subset of A_N is externally stable. We can guess a weak saddle S containing $d^* \in S$ in nondeterministic polynomial time and verify its minimality by checking that none of its subsets are externally stable. This places INWEAKSADDLE in NP^{coNP} and thus in Σ_2^p. □

5 Very Weak Saddles

A natural weakening of weak dominance is *very weak dominance*, which does not require a strict inequality in addition to the weak inequalities [14]. Thus, in particular, two actions that always yield the same payoff very weakly dominate each other. Formally, for a player $i \in N$ and two actions $a_i, b_i \in A_i$ we say that a_i

very weakly dominates b_i with respect to S_{-i}, denoted $a_i \geq_{S_{-i}} b_i$, if $p_i(a_i, s_{-i}) \geq p_i(b_i, s_{-i})$ for all $s_{-i} \in S_{-i}$. Based on this modified notion of dominance, one can define the very weak analog of the weak saddle (cf. Definition 2).

Computational problems for very weak saddles are defined analogously to their counterparts for weak saddles. It turns out that most of our results for the latter can be transferred to the former.

Theorem 4. *The following holds:*

- InVeryWeakSaddle *is NP-hard.*
- InVeryWeakSaddle *is coNP-hard.*
- IsVeryWeakSaddle *is coNP-complete.*
- InAllVeryWeakSaddles *is coNP-complete.*
- UniqueVeryWeakSaddle *is coNP-hard.*
- NontrivialVeryWeakSaddle *is NP-complete.*
- FindVeryWeakSaddle *is NP-hard.*

It should be noted that the hardness results for very weak saddles do not follow in an obvious way from the corresponding results for weak saddles, or vice versa. While the proofs are based on the same general idea, and again on one core construction, there are some significant technical differences.

6 Conclusion

In the early 1950s, Shapley proposed an ordinal set-valued solution concept known as the weak saddle. We have shown that weak saddles are intractable in bimatrix games. As it turned out, not only *finding* but also *recognizing* weak saddles is computationally hard. This distinguishes weak saddles from Nash equilibrium, iterated dominance, and any other game-theoretic solution concept we are aware of. Three of the most challenging remaining problems are to study the complexity of weak saddles in matrix games, to close the gap between Θ_2^p and Σ_2^p for InWeakSaddle, and to completely characterize the complexity of FindWeakSaddle.

Acknowledgements. This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR-2312/6-1 and BR 2312/3-2.

References

1. Basu, K., Weibull, J.: Strategy subsets closed under rational behavior. Economics Letters 36, 141–146 (1991)
2. Baumeister, D., Brandt, F., Fischer, F., Hoffmann, J., Rothe, J.: The complexity of computing minimal unidirectional covering sets. Technical report (2009), http://arxiv.org/abs/0901.3692
3. Bernheim, B.: Rationalizable strategic behavior. Econometrica 52(4), 1007–1028 (1984)

4. Brandt, F., Fischer, F.: Computing the minimal covering set. Mathematical Social Sciences 56(2), 254–268 (2008)
5. Brandt, F., Brill, M., Fischer, F., Harrenstein, P.: Computational aspects of Shapley's saddles. In: Proc. of 8th AAMAS Conference, pp. 209–216 (2009)
6. Brandt, F., Brill, M., Fischer, F., Harrenstein, P.: On the complexity of iterated weak dominance in constant-sum games. In: Mavronicolas, M., Papadopoulou, V.G. (eds.) SAGT 2009. LNCS, vol. 5814, pp. 287–298. Springer, Heidelberg (2009)
7. Chen, X., Deng, X.: Settling the complexity of 2-player Nash-equilibrium. In: Proc. of 47th FOCS Symposium, pp. 261–272. IEEE Press, Los Alamitos (2006)
8. Conitzer, V., Sandholm, T.: Complexity of (iterated) dominance. In: Proc. of 6th ACM-EC Conference, pp. 88–97. ACM Press, New York (2005)
9. Daskalakis, C., Goldberg, P., Papadimitriou, C.: The complexity of computing a Nash equilibrium. In: Proc. of 38th STOC, pp. 71–78. ACM Press, New York (2006)
10. Duggan, J., Le Breton, M.: Dutta's minimal covering set and Shapley's saddles. Journal of Economic Theory 70, 257–265 (1996)
11. Dutta, B.: Covering sets and a new Condorcet choice correspondence. Journal of Economic Theory 44, 63–80 (1988)
12. Gilboa, I., Kalai, E., Zemel, E.: The complexity of eliminating dominated strategies. Mathematics of Operations Research 18(3), 553–565 (1993)
13. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. Journal of the ACM 44(6), 806–825 (1997)
14. Luce, R.D., Raiffa, H.: Games and Decisions: Introduction and Critical Survey. John Wiley & Sons Inc., Chichester (1957)
15. McKelvey, R.D., Ordeshook, P.C.: Symmetric spatial games without majority rule equilibria. The American Political Science Review 70(4), 1172–1184 (1976)
16. Myerson, R.B.: Game Theory: Analysis of Conflict. Harvard University Press, Cambridge (1991)
17. Nash, J.F.: Non-cooperative games. Annals of Mathematics 54(2), 286–295 (1951)
18. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
19. Pearce, D.: Rationalizable strategic behavior and the problem of perfection. Econometrica 52(4), 1029–1050 (1984)
20. Samuelson, L.: Dominated strategies and common knowledge. Games and Economic Behavior 4, 284–313 (1992)
21. Shapley, L.: Order matrices. I. Technical Report RM-1142, The RAND Corporation (1953)
22. Shapley, L.: Order matrices. II. Technical Report RM-1145, The RAND Corporation (1953)
23. Shapley, L.: Some topics in two-person games. In: Dresher, M., Shapley, L.S., Tucker, A.W. (eds.) Advances in Game Theory. Annals of Mathematics Studies, vol. 52, pp. 1–29. Princeton University Press, Princeton (1964)
24. von Neumann, J.: Zur Theorie der Gesellschaftspiele. Mathematische Annalen 100, 295–320 (1928)
25. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)
26. Wagner, K.: More complicated questions about maxima and minima, and some closures of NP. Theoretical Computer Science 51, 53–80 (1987)

Learning and Approximating the Optimal Strategy to Commit To*

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala

Department of Computer Science, Duke University, Durham, NC, USA
{jcl,conitzer,kamesh}@cs.duke.edu

Abstract. Computing optimal Stackelberg strategies in general two-player Bayesian games (not to be confused with Stackelberg strategies in routing games) is a topic that has recently been gaining attention, due to their application in various security and law enforcement scenarios. Earlier results consider the computation of optimal Stackelberg strategies, given that all the payoffs and the prior distribution over types are known. We extend these results in two different ways. First, we consider *learning* optimal Stackelberg strategies. Our results here are mostly positive. Second, we consider computing *approximately* optimal Stackelberg strategies. Our results here are mostly negative.

1 Introduction

Game theory defines solution concepts for strategic situations, in which multiple self-interested agents interact in the same environment. Perhaps the best-known solution concept is that of *Nash equilibrium* [11]. A Nash equilibrium prescribes a strategy for every player, in such a way that no individual player has an incentive to change her strategy. If strategies are allowed to be mixed—a mixed strategy is a probability distribution over pure strategies—then it is known that every finite game has at least one Nash equilibrium. Some games have more than one equilibrium, leading to the *equilibrium selection problem*.

Perhaps the most basic representation of a game is the *normal form*. In the normal-form representation, every player's pure strategies are explicitly listed, and for every combination of pure strategies, every player's utility is explicitly listed.

The problem of *computing* Nash equilibria of a normal-form game has received a large amount of attention in recent years. Finding a Nash equilibrium is PPAD-complete [6,1]. Finding an optimal equilibrium (for just about any reasonable definition of "optimal"—for instance, maximizing the sum of the players' utilities) is NP-hard [7,3]; moreover, it is not even possible to find an equilibrium that is approximately optimal in polynomial time, unless P=NP [3]. This holds even for two-player games. However, Nash equilibrium is not always the right solution concept. In some settings, one player can credibly commit to a strategy, and communicate this to the other player, before the other player can make a decision. To see how this can affect the outcome of

* Some of these results were briefly presented as part of a talk at the 2009 Bellairs Workshop on Algorithmic Game Theory. This work is funded by: Alfred P. Sloan Research Fellowships, NSF Grant IIS-0812113, NSF Career Award 0745761 and Grant CNS-0540347.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 250–262, 2009.

	L	R
U	(2,1)	(4,0)
D	(1,0)	(3,1)

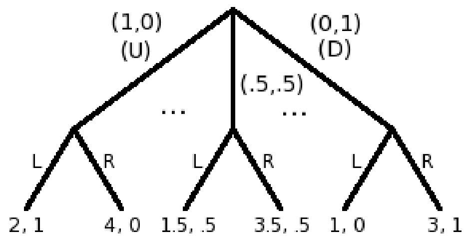


Fig. 1. A sample game and its extensive form representation

a game, consider the following simple normal-form game (which has previously been used as an example for this, *e.g.*, [2]):

For the case where the players move simultaneously (no ability to commit), the unique Nash equilibrium is (U, L): U strictly dominates D, so that the game is solvable by iterated strict dominance. So, player 1 (the row player) receives utility 2. However, now suppose that player 1 has the ability to commit. Then, she is better off committing to play D, which will incentivize player 2 to play R, resulting in a utility of 3 for player 1. The situation gets even better for player 1 if she can commit to a mixed strategy: in this case, she can commit to the mixed strategy $(.5 - \epsilon, .5 + \epsilon)$, which still incentivizes player 2 to play R, but now player 1 receives an expected utility of $3.5 - \epsilon$. To ensure the existence of optimal strategies, we assume (as is commonly done [2,12]) that player 2 breaks ties in player 1's favor, so that the optimal strategy for player 1 to commit to is $(.5, .5)$, resulting in a utility of 3.5. (Note that there is never a reason for player 2 to randomize, since he effectively faces a single-agent decision problem.) An optimal strategy to commit to is usually called a *Stackelberg* strategy, after von Stackelberg, who showed that in Cournot's duopoly model [4], a firm that can commit to a production quantity has a strategic advantage [15]. Throughout this paper, a Stackelberg strategy is an optimal *mixed* strategy to commit to; we will only consider two-player games. In this context, the Stackelberg leader's expected utility is always at least the expected utility that she would receive in any Nash (or even correlated) equilibrium of the simultaneous-move game [16]. In contrast, committing to a pure strategy is not always beneficial; for example, consider matching pennies.

One may argue that the normal form is not the correct representation for this game. In game theory, the time structure of games is usually represented by the *extensive form*. Indeed, the above game can be represented as the extensive-form game in Figure 1. While this is a conceptually useful representation, from a computational perspective it is not helpful: player 1 has an infinite number of strategies, hence (the naïve representation of) the tree has infinite size. It should be emphasized that committing to a mixed strategy is *not* the same as randomizing over which pure strategy to commit to; in fact, there is no reason to randomize over which strategy to commit to. Thus, from a computational viewpoint, it makes more sense to operate directly on the normal form.

The problem of computing Stackelberg strategies in general normal-form (or, more generally, Bayesian) games has only recently started to receive attention. A 2006 EC paper by Conitzer and Sandholm [2] layed out the basic complexity results for this setting: Stackelberg strategies can be computed in polynomial time for two-player general-sum normal-form games using linear programming (in contrast to the problem of finding a Nash equilibrium), but computing Stackelberg strategies is NP-hard for two-player Bayesian games or three-player normal-form games. Undeterred by the NP-hardness result, Paruchuri *et al.* [12] developed a mixed-integer program for finding an (optimal) Stackelberg strategy in the two-player Bayesian case (the setting that we study in this paper). They show that using this formulation is much faster than converting the game to normal form (leading to an exponential increase in size) and then using the linear programming approach. Moreover, this algorithm forms the basis for their deployed ARMOR system, which is used at the Los Angeles International Airport to randomly place checkpoints on roads entering the airport, as well as to decide on canine patrol routes [9,13]. The use of commitment in similar games dates back much further, including, for example, applications to inspection games [10]. The formal properties of various types of commitment are also studied in [8].

It should be noted that Stackelberg strategies are a generalization of minimax strategies in two-player zero-sum games. Because computing minimax strategies is equivalent to linear programming [5], this also implies that a linear programming solution for computing Stackelberg strategies is the best that we can hope for. Of course, Nash equilibrium is an alternative generalization of minimax strategies. Stackelberg strategies have the significant advantage that they avoid the equilibrium selection problem: there is an optimal value of the game for the leader (player 1), which in general corresponds to a single optimal strategy (though not in degenerate cases). The notion of "Stackelberg strategies" has appeared in other contexts in the algorithmic game theory literature, specifically, in the context of routing games, where a single benevolent party controls part of the flow, and commits to routing this flow in a manner that minimizes total latency [14]. While interesting, that paper does not seem that closely related to our work, because in our context, the leader is a selfish player in an arbitrary game.

The rest of this paper is layed out as follows. In Section 2, we formally review the necessary concepts, introduce our notation, and discuss existing results that are relevant. In Section 3—the first half of our contribution—we prove several results about *learning* Stackelberg strategies, in contexts where the follower payoffs and/or the distribution over types is not known initially. In Section 4—the second half of our contribution—we consider purely computational problems and give (in)approximability results.

2 Preliminaries

In this section, we review notation and existing results.

2.1 Notation and Definitions

We will refer to player 1 as the *leader* and to player 2 as the *follower*. Let A_l be the set of leader actions in the game ($|A_l| = d$), and let A_f be the set of follower actions ($|A_f| = k$). The leader's utility is given by a function $u_l : A_l \times A_f \rightarrow \mathbb{R}$. When we are studying approximability, we (wlog) require all the leader utilities to be nonnegative (to make multiplicative approximation meaningful). In a Bayesian game, the follower has a set of *types* Θ ($|\Theta| = \tau$), which, together with the actions taken, determine his utility, according to a function $u_f : \Theta \times A_l \times A_f \to \mathbb{R}$. For simplicity, we will not consider situations where the leader's utility also depends on the follower's type; this restriction strengthens our hardness results. We will refer to these as *Bayesian* games; a *normal-form* game is the special case where there is only a single type.

σ denotes a mixed strategy for the leader, and $\sigma(a_l)$ the probability that σ places on action a_l. Let $\text{BR}(\theta, \sigma) \in A_f$ denote the action that the follower plays (that is, his best response, with ties broken in favor of the leader) when his type is θ and the leader has committed to playing σ. We note that

$$\text{BR}(\theta, \sigma) \in \arg\max_{a_f \in A_f} \sum_{a_l \in A_l} \sigma(a_l) u_f(\theta, a_l, a_f)$$

The BR function also captures the fact that the follower breaks ties in the leader's favor. Given the follower type θ, the leader's expected utility is

$$\sum_{a_l \in A_l} \sigma(a_l) u_l(a_l, \text{BR}(\theta, \sigma))$$

Given a prior probability distribution $P : \Theta \to [0, 1]$ over follower types, the leader's expected utility for committing to σ is

$$\sum_{\theta \in \Theta} P(\theta) \sum_{a_l \in A_l} \sigma(a_l) u_l(a_l, \text{BR}(\theta, \sigma))$$

When we take a worst-case perspective, we will be interested in a setting with types but without a prior distribution over them (also known as a *pre-Bayesian* game).

2.2 Known Results and Techniques

In this subsection we review the most relevant prior work. For a normal-form game, the optimal mixed leader strategy can be computed in polynomial time, as follows:[1] for every follower action a_f, the following linear program (whose variables are the $\sigma(a_l)$) can be used to determine the best leader strategy that makes the follower play a_f:

maximize $\sum_{a_l} \sigma(a_l) u_l(a_l, a_f)$
subject to
$(\forall a'_f)\ \sum_{a_l} \sigma(a_l) u_f(a_l, a_f) \geq \sum_{a_l} \sigma(a_l) u_f(a_l, a'_f)$
$\sum_{a_l} \sigma(a_l) = 1$
$(\forall a_l)\ \sigma(a_l) \geq 0$

Some of these linear programs may be infeasible (it is impossible to make a follower play a strictly dominated strategy), but some will be feasible; the solution of the one with the highest objective value gives the optimal mixed strategy for the leader.

For Bayesian games (with a prior), the problem of computing the optimal mixed leader strategy is known to be NP-hard [2]. However, this strategy can be found using a mixed integer program [12].

[1] This algorithm was presented in [2]. Some of the analysis in [16] is based on similar insights.

2.3 Visualization

In this subsection, we show how the problems we discussed above can be visualized. Let us consider the normal-form case. The space of possible strategies for the leader defines a unit simplex in $d-1$ dimensions, where d is the number of leader actions. For each strategy of the leader, the follower has a best response. The space of leader strategies for which the follower's best response is a_f defines a (possibly empty) polyhedron. Therefore, the d-simplex splits into at most k (number of follower actions) polyhedral regions, based on the follower utility function. Each of these regions corresponds to the feasible region of one of the linear programs, and the objective of that linear program can be represented as an arrow in the region.

Let us consider the following small example and its visualization.

	L	C	R
U	(0,1)	(1,0)	(0,0)
M	(4,0)	(0,1)	(0,0)
D	(0,0)	(1,0)	(1,1)

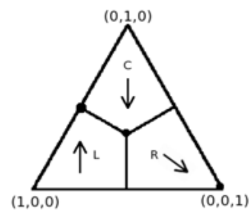


Fig. 2. A small game and its visualization

Each dot in Figure 2 represents the optimal point (leader mixed strategy) within each region (which lie on *separating hyperplanes* or on the boundary); the largest dot (.5,.5,0) shows the optimal point overall.

The Bayesian case can be visualized in (at least) two different ways. A simple way is to have a separate unit simplex for every type; this does not require a prior distribution over types (that is, it works for pre-Bayesian games). If there is a prior distribution over types, another way is to have a region for each element of the set of all pure strategies for the follower, so that $(a_f^{\theta^1}, \ldots, a_f^{\theta^T})$ corresponds to the region where type θ^1's best response is $a_f^{\theta^1}$, type θ^2's best response is $a_f^{\theta^2}$, *etc.* The arrows in this region represent the objective, which depends on the prior. This representation does not work for pre-Bayesian games where we take a worst-case perspective, because the optimal point may be in the interior of a region.

3 Learning Stackelberg Strategies

If a game is repeated over time, this opens up the possibility for the leader to learn something about the follower's utilities or the distribution over types. To avoid the possibility that the follower tries to mislead the leader over time, we imagine that a new follower agent is drawn in every round. Alternatively, the follower can be assumed to behave myopically. In a round, the leader commits to a mixed strategy, and subsequently observes the follower's response. The leader's goal is to learn enough to determine the optimal Stackelberg strategy, in as few rounds (*samples*) as possible.

Due to space constraint, we focus on the case with a single type: that is, in each round, the follower has the same payoff matrix, given by $u_f(a_l, a_f)$, initially unknown to the leader. In each round, the leader commits to a mixed strategy σ and learns the follower's response. We say that the leader *queries* or *samples* the point σ on the probability simplex. The goal is to minimize the number of samples necessary to find the optimal (Stackelberg) mixed strategy for the leader. In the full version of this paper (Appendices B,C) we consider two other cases with more than one type, one where the leader needs to learn the follower payoff function, and one where this function is known, but the leader must discover the distribution over types. We make the following assumptions:

- The follower utilities are non-degenerate; no separating hyperplanes coincide.
- We will only consider regions whose volume is at least some fraction $\epsilon > 0$ of the total volume, and try to find the optimal solution among points in these regions. (It can be argued that solutions in smaller regions are too unstable. Alternatively, we can simply assume that every nonempty region has at least this volume.)
- We assume that the optimal solution can be specified exactly using a limited amount of precision quantified by L. This allows us to bound the number of iterations of binary search needed to calculate these hyperplanes exactly, to a linear multiple of L.

Our approach will be to learn all the regions (whose volume is at least ϵ of the total)—that is, find all hyperplanes separating these regions. Once we know these, the optimal strategy can be computed using the linear programming approach above.

A high-level outline of our algorithm SU is as follows. For each follower action $a_f \in A_f$, the algorithm maintains an overestimate P_{a_f} of the region where a_f is a best response. It then refines these overestimates via sampling, until they are disjoint.

SU

1. For each $a_f \in A_f$, find a point (leader strategy) q_{a_f} in the d-simplex to which a_f is a best response (provided the corresponding region is sufficiently large).
2. Initially, each P_{a_f} is the entire d-simplex.
3. Repeat the following until all P_{a_f} are disjoint:
   (a) Find a point p^* in the intersection of some $P_{a'_f}$ and $P_{a''_f}$.
   (b) Sample to obtain the optimal follower strategy at p^*; call it a^*_f.
   (c) Draw a line segment between p^* and some q_{a_f} for $a_f \neq a^*_f, a_f \in \{a'_f, a''_f\}$; perform binary search on this line to find a single point on a hyperplane that we have not yet discovered.
   (d) Find a set of d linearly independent points on the hyperplane, and hence reconstruct it.
   (e) Update the P_{a_f} to take this new hyperplane into account.

We now describe the steps of SU in detail.

Step (1). Finding a point in each region (with at least ϵ of the volume) can be achieved via random sampling, via the following lemma.

Lemma 1. *It takes $O(Fk\log k)$ samples to w.h.p. (with high probability) find a single point in each sufficiently large region, where $F = 1/\epsilon$.*

Proof. The probability that a randomly chosen point corresponds to follower action a_f is at least ϵ. Therefore, for any constant integer $c \geq 1$, after $((c+1)F\log k)$ samples, the probability that follower action a_f is not hit is at most $(\frac{1}{k})^{c+1}$. By a union bound, the probability that at least one action is not hit is at most $(\frac{1}{k})^{c}$.

Step (3 a–c). Consider two overestimates $P_{a'_f}$ and $P_{a''_f}$ that have nonzero overlap volume. By Step (1), we may assume that we have sampled a point $q_{a'_f}$ that led to a response of a'_f (that is, $q_{a'_f}$ is in the region corresponding to a'_f), and a point $q_{a''_f}$ that led to a response of a''_f. Both of these overestimates are characterized by sets H' and H'' of hyperplanes that we have previously discovered. We need to discover a new hyperplane. It will not suffice to do binary search on the line segment between the two starting points, as illustrated by Figure 3, which illustrates a situation where we have discovered two of the hyperplanes of Figure 2. If we do binary search on the line segment between the two indicated points, we cannot discover the missing hyperplane, because the top region "gets in the way" (another action, namely C, will start being the best response). However, if we sample from the shaded set $P_L \cap P_R$, the result will be different from one of the two points; then, by performing binary search on the line segment between this point and the new point, we will find a point on a new hyperplane. The following algorithm formalizes this idea. In it, we do not assume that the two overestimates overlap

FIND POINT

1. Solve a linear program to find an interior point p^* of $P_{a'_f} \cap P_{a''_f}$ given the constraints $H' \cup H''$. (If this is not feasible, return failure.)
2. Sample this point and let the follower strategy returned be a^*_f.
   (a) If $a^*_f = a'_f$, search the line segment between p^* and $q_{a''_f}$ for a point on a hyperplane that has the region corresponding to a''_f adjacent on one side, via binary search.
   (b) Otherwise, search the line segment between p^* and $q_{a'_f}$ for a point on a hyperplane that has the region corresponding to a'_f adjacent on one side, via binary search.

Lemma 2. *Given overestimates $P_{a'_f}$ and $P_{a''_f}$ on the regions corresponding to a'_f and a''_f, and points $q_{a'_f}$ and $q_{a''_f}$ in these respective regions,* FIND POINT *will either give a point on a new hyperplane for one of the regions $P_{a'_f}$ or $P_{a''_f}$, or will return that $P_{a'_f}$ and $P_{a''_f}$ already have zero intersection volume. This requires $O(L)$ samples.*

The detailed proof is in Appendix A of the full paper.

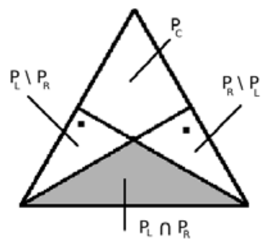


Fig. 3. Finding a hyperplane

Step (3d). In this step, the input is a point p on the hyperplane that we need to reconstruct, and the two follower actions a'_f and a''_f that correspond to the regions separated by this hyperplane. The following DETERMINE HYPERPLANE finds the hyperplane.

DETERMINE HYPERPLANE

1. Sample the vertices of a regular d-simplex with sides of length $\epsilon' \ll \epsilon$, centered at p. (Draw this simplex uniformly at random among such simplices.)
2. Organize the vertices of this simplex into two sets, V' and V'' according to the region they fall in. (Both of these sets will be nonempty.)
3. Choose d distinct pairs of points where one of the points is in V' and the other is in V''
4. Binary-search the d line segments formed by these pairs, to find the points where these line segments intersect the hyperplane.

Lemma 3. DETERMINE HYPERPLANE *will give d linearly independent points on the hyperplane using $O(dL)$ samples.*

Proof. First, consider the $d + 1$ vertices of the d-simplex centered at p. Since ϵ' is sufficiently small, all of the points fall into one of the two regions (and since the simplex is chosen at random, there is zero probability of one of the vertices being exactly on the hyperplane). Since the hyperplane goes through p, at least one of the vertices of the simplex will fall into each region. As a result, there are at least d line segments between vertices of the simplex where the two vertices of the segment produce different follower actions. Finally, the points where the hyperplane intersects with these line segments must be linearly independent; otherwise, the simplex would not be full-dimensional. Furthermore, the number of samples needed to find the hyperplane-intersecting point on a line segment via binary search is linear in L. This completes the proof.

With these tools, we can give our main result for this problem:

Theorem 1. *To find, w.h.p., all the hyperplanes that separate regions, SU requires $O(Fk \log k + dk^2 L))$ samples, where $F = 1/\epsilon$, ϵ is the smallest volume of regions that we consider, L is the precision, and $k = |A_f|$. Computationally, this requires the solution of $O(k^2)$ linear programs.*

Details of the proof are in Appendix A of the full paper. Once we have generated all the hyperplanes that separate regions, we can use the known linear programming approach described in Subsection 2.2 to find the optimal mixed strategy to commit to.

4 Computing Stackelberg Strategies

In this section, we consider how different modeling assumptions affect the computational tractability and approximability of the Stackelberg problem with multiple follower types. Unlike the previous section, this section does not consider learning problems at all: it focuses strictly on the computational aspects of the optimization. Because of this, we only consider a single-round setting in this section.

The following aspects of the model will remain the same throughout this section.

- We consider two-player, general-sum games that have more than one follower type.
- The leader's utility does not depend *directly* on the follower's type (but it does depend on the follower's action, which can be affected by the follower's type).
- The follower's utility function $u_f(\theta, a_l, a_f)$ is common knowledge.

We consider two modeling decisions. The first decision concerns whether the type space is discrete or continuous. For the discrete case, we assume that we have a finite number of types, which are explicitly listed. For the continuous case, we assume that the space of possible types is defined by a lower bound and an upper bound for the follower's utility for each action profile (a_l, a_f); every follower payoff matrix that is consistent with these bounds corresponds to some type.

The second modeling decision is whether the follower type is chosen according to a Bayesian model or an adversarial (worst-case) model. Note that the "adversary" is *not* one of the players of the game, in particular, the adversary and the follower are different.

4.1 Computing Bayesian Optimal Strategies with Finitely Many Types

In this subsection we study how to compute the optimal mixed strategy when the follower's type is drawn from a known distribution over finitely many types. We refer to this problem as *Bayesian optimization for finite types (BOFT)*. BOFT is defined as:

- We have a set Θ of possible follower types, $|\Theta| = \tau$.
- The follower's utility function $u_f(\theta, a_l, a_f)$ is common knowledge.
- Both the follower's utility function $u_f(\theta, a_l, a_f)$ and the leader's utility function $u_f(\theta, a_l, a_f)$ are normalized to lie in [0,1] for all inputs.
- The prior over follower types $P(\theta)$ is common knowledge.
- An optimal leader strategy is one that maximizes the leader's expected utility.

This problem was first studied in [2], where it was shown to be NP-hard. It also forms the basis for much of the applied work on computing Stackelberg strategies [9]. However, to the best of our knowledge, the approximability of this problem has not yet been studied. We settle the approximability precisely in this subsection.

Theorem 2. *For all constant $\epsilon > 0$, no polynomial-time factor-$\tau^{1-\epsilon}$ approximation exists for BOFT unless NP = P, even if there are only two follower actions.*

This hardness of approximation can be shown by a reduction from MAX-INDEPENDENT-SET. In this reduction, vertices correspond to types, and the leader cannot incentivize two adjacent types to both play a desirable action. The full reduction appears in Appendix D of the full paper.

Theorem 3. *There is a polynomial-time factor-τ approximation algorithm for BOFT.*

A simple algorithm that achieves this is the following: choose a type uniformly at random, and solve for the optimal mixed strategy to commit to for this specific type (using the linear programming approach). With probability $1/\tau$, we choose the type that is actually realized, in which case we perform at least as well as the optimal overall strategy. Hence, this guarantees at least a τ approximation. Details and derandomization appear in Appendix D of the full paper.

4.2 Computing Worst-Case Optimal Strategies with Finitely Many Types

A prior distribution over follower types is not always readily available. In that case, we may wish to optimize for the worst-case type (equivalently, the worst-case distribution over types). We note that the worst-case type depends on the mixed strategy that we choose, so that this is not the same problem as optimizing against a single type. We refer to this problem as *worst-case optimization for finite types (WOFT)*:

- We have a set Θ of possible follower types, $|\Theta| = \tau$.
- The follower's utility function $u_f(\theta, a_l, a_f)$ is common knowledge.
- An optimal leader strategy is one that maximizes the worst-case expected utility for the leader, where the worst case is taken over follower types (but we are taking the expectation over the mixed strategy). That is, an adversary (not equal to the follower) chooses the follower type after the leader mixed strategy is chosen, but before the pure-strategy realization.

It turns out that WOFT is even less approximable than BOFT.

Theorem 4. *WOFT is completely inapproximable in polynomial time, unless P=NP (that is, it is hard to distinguish between instances where the leader can get at least* 1 *in the worst case, and instances where the leader can only get* 0*)—even if there are only four follower actions.*

This can be shown by a reduction from 3-SAT. In the resulting game, the leader can obtain an expected utility of 1 against every type if the 3-SAT instance is satisfiable, and otherwise will obtain utility 0 against some type. The full reduction appears in Appendix D of the full paper.

4.3 Optimizing for the Worst Type with Ranges

So far, we have assumed that the space of possible types is represented by explicitly listing the (finitely many) types and the corresponding utilities. However, this representation of the uncertainty that the leader has over the follower's preferences is not always convenient. For example, the leader may have a rough idea of every follower payoff, which could be represented by a range in which that payoff must lie. This corresponds to a continuous type space for the follower: every setting of all the follower payoffs within the ranges corresponds to a type.

In this subsection, we study the problem of maximizing the leader's worst-case utility over all types (instantiations of the follower payoffs within the ranges). Later in the subsection, we also consider a generalization where the follower payoffs in different entries can be linked to each other.

For example, consider the following game with ranges:

	L	R
U	0, [1,2]	1, 0
D	1,0	0, [1,2]

The leader is unsure about the follower's utility for (U, L) and (D, R), each of which is known to lie somewhere in the range $[1, 2]$ (they can vary independently). The follower knows his utilities. If the leader places less than $1/3$ probability on U, then the follower is guaranteed to play R; this results in a utility of at most $1/3$ for the leader. If the leader places more than $2/3$ probability on U, then the follower is guaranteed to play L; this results in a utility of at most $1/3$ for the leader. If the leader places probability between $1/3$ and $2/3$ on U, then the follower may end up playing either L or R; by placing probability $1/2$ on U, the leader obtains an expected utility of $1/2$, which is optimal.

We refer to this problem as *worst-case optimization for range types (WORT)*:

- For every (a_l, a_f), the leader has a range in which the follower utility might lie, $u_f(a_l, a_f) \in [u_f^l(a_l, a_f), u_f^h(a_l, a_f)]$. The leader knows her own utilities $u_l(a_l, a_f)$.
- An optimal leader strategy is one that maximizes the worst-case expected utility for the leader, where the worst-case values of

Theorem 5. *WORT is NP-hard.*

This follows from a reduction from 3-COVER, which is presented in Appendix D of the full paper. It is an open question whether WORT can be efficiently approximated. In Appendix E of the full paper, we define a generalization of WORT, which we prove is inapproximable unless $P = NP$. This generalization allows the follower's payoffs to be linked across entries.

5 Conclusion

Computing optimal Stackelberg strategies in general two-player Bayesian games is a topic that has been gaining attention in recent years, due to their application in both security and law enforcement. Earlier results consider the computation of optimal Stackelberg strategies, given that all the payoffs and the prior distribution over types are known. We extended these results in two ways.

First, we considered *learning* optimal Stackelberg strategies. We first considered the normal-form case where the follower payoffs are not known and showed how we can efficiently learn enough about the payoffs to determine the optimal strategy. We then extended this to Bayesian games. We also considered the case where the payoffs are known, but the distribution over types is not. We showed how we can efficiently learn enough about the distribution to determine the optimal strategy. It must be admitted that it is debatable whether this framework for learning is practical for current real-world security applications, since the costs incurred during the learning phase may be too high; however, these costs may be more manageable in electronic commerce applications.

Second, we considered computing *approximately* optimal Stackelberg strategies. Our results here were mostly negative: we showed that the best possible approximation ratio that can be obtained in polynomial time for the standard Bayesian problem is τ, the number of types, unless NP = P. Optimizing for the worst type is completely inapproximable in polynomial time, in the sense that we cannot distinguish instances where we can guarantee utility 1 from instances where it is impossible to guarantee positive utility, unless P=NP. We also studied a different representation of uncertainty about the follower's payoffs that relies on ranges, and showed that optimizing for the worst case is NP-hard in the basic setting, and completely inapproximable in a generalized setting where the payoffs are linked. These negative results provide some justification for the use of worst-case exponential-time algorithms in this context, such as those that use mixed integer programming.

Two immediate directions for future research are: (1) investigating the approximability of the basic ranges problem, and (2) considering the ranges problem in the Bayesian case (rather than the worst case). There are many other directions for future research, for example, studying the number of samples required to learn *approximately* optimal strategies, investigating the case where there are more than two players, and/or computing optimal Stackelberg strategies when the normal form has exponential size, but the game is concisely represented.

References

1. Chen, X., Deng, X.: Settling the complexity of two-player Nash equilibrium. In: FOCS, pp. 261–272 (2006)
2. Conitzer, V., Sandholm, T.: Computing the optimal strategy to commit to. In: Proceedings of the ACM Conference of EC, Ann Arbor, MI, USA, pp. 82–90 (2006)
3. Conitzer, V., Sandholm, T.: New complexity results about Nash equilibria. Games and Economic Behavior 63(2), 621–641 (2008)
4. Cournot, A.A.: Recherches sur les principes mathématiques de la théorie des richesses (Researches into the Mathematical Principles of the Theory of Wealth) (1838)
5. Dantzig, G.: A proof of the equivalence of the programming problem and the game problem. In: Koopmans, T. (ed.) Activity Analysis of Production and Allocation, pp. 330–335. John Wiley & Sons, Chichester (1951)
6. Daskalakis, C., Goldberg, P., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. In: STOC, pp. 71–78 (2006)
7. Gilboa, I., Zemel, E.: Nash and correlated equilibria: Some complexity considerations. Games and Economic Behavior 1, 80–93 (1989)
8. Harrenstein, P., Brandt, F., Fischer, F.: Commitment and extortion. In: Proceedings of AAMAS, Honolulu, HI, USA (2007)

9. Jain, M., Pita, J., Tambe, M., Ordóñez, F.: Bayesian Stackelberg games and their application for security at Los Angeles international airport. SIGecom Exch. 7(2), 1–3 (2008)
10. Maschler, M.: A price leadership method for solving the inspector's non-constant-sum game. Naval Research Logistics Quarterly 13(1), 11–33 (1966)
11. Nash, J.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 36, 48–49 (1950)
12. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordóñez, F.: Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. In: Proceedings of AAMAS, Estoril, Portugal, pp. 895–902 (2008)
13. Pita, J., Jain, M., Ordóñez, F., Portway, C., Tambe, M., Western, C.: Using game theory for Los Angeles airport security. AI Mag. 30(1), 43–57 (2009)
14. Roughgarden, T.: Stackelberg scheduling strategies. In: STOC, pp. 104–113. ACM, New York (2001)
15. von Stackelberg, H.: Marktform und Gleichgewicht. Springer, Vienna (1934)
16. von Stengel, B., Zamir, S.: Leadership with commitment to mixed strategies. Research Report LSE-CDAM-2004-01, London School of Economics (February 2004)
17. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. Theory of Computing 3(1), 103–128 (2007)

Doing Good with Spam Is Hard*

Martin Hoefer, Lars Olbrich, and Alexander Skopalik

Department of Computer Science, RWTH Aachen University, Germany

Abstract. We study economic means to improve network performance in the well-known game theoretic traffic model due to Wardrop. We introduce two sorts of spam flow - auxiliary and adversarial flow - that have no intrinsic value. Auxiliary/adversarial flows are a separate commodity with the sole objective to minimize/maximize the latency at the induced Wardrop equilibrium of the selfish flow. By this means a separate access to the edges is not necessary and the latency of the regulating flow does not distort the arising latency cost. We present networks where auxiliary flow is able to improve the network performance. However, we show that the optimal auxiliary flow is NP-hard to compute and not approximable within a factor of less then $\frac{4}{3}$. The minimal amount of auxiliary flow needed to induce the best possible equilibrium is even hard to approximate by any subexponential factor. These hardness results are complemented by proving NP-hardness for the optimal adversarial flow. All hardness results hold even for single-commodity networks.

1 Introduction

Wardrop's traffic model is a well-studied model for routing with important applications in road traffic and computer networks. In this model, we are given a network equipped with non-decreasing non-negative latency functions mapping flow on the edges to latency. For each of several commodities a fixed demand has to be routed between a source-sink pair. The cost of a flow assignment is the weighted sum of travel times between the source and target nodes. A flow that minimizes the total latency is called *(socially) optimal*. A common interpretation of the Wardrop model is that flow is controlled by an infinite number of selfish users each of which carries an infinitesimal amount of flow. Each user aims at minimizing its path latency. An allocation in which no user can improve its situation by unilaterally deviating from its current path is called *Wardrop equilibrium*. In general a Wardrop equilibrium is not socially optimal, i.e, it does not minimize the total latency. The inefficiency of selfish flows has been extensively studied in previous work [3, 23, 24, 26].

We study a means of reducing the inefficiency of selfish flow applicable in scenarios with no central control. There have been several approaches to this problem in the literature, most prominently taxing, Stackelberg routing, and

* Supported by the DFG GK/1298 "AlgoSyn", by the German Israeli Foundation (GIF) under contract 877/05, and by DFG through UMIC Research Centre at RWTH Aachen University.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 263–274, 2009.

network design, but there are some problems with these approaches in large networks without strong centralized control. Taxing requires to collect possibly different taxes at each edge, a process that requires an infrastructure that can be costly or impossible to establish. In addition, a look at classical taxing procedures from a user perspective reveals that, albeit taxes improve the latency of the networks, they do not improve the disutility of users for a large set of networks [7]. In Stackelberg routing the idea is to put a fraction of selfish flow under centralized control and reroute this flow such that the total latency of all flow is optimized. Here the underlying assumption that a central control agency can directly manipulate the selfish demand is quite strong. Finally, network design requires to manipulate the network structure, which is clearly a strong assumption of centralized control in a large network.

In this paper, we consider a means of control motivated by the concept of spam in the Internet. We introduce two sorts of *spam flows*, which we call auxiliary and adversarial flow. The demand value of these flows is given independently in addition to the given selfish flow demand. Spam flow can be seen as a separate altruistic or malicious commodity that tries to influence the routing decisions of selfish players without directly taking control over (parts of) the players or the network. The goal is to route the spam flow in such a way that the induced equilibrium minimizes/maximizes the total latency of the selfish flow. The routed packets solely alter the latency of the used edges. They have no value and are essentially spam. Therefore we assume that the latency of spam flow does not contribute to the social cost.

Our results. We first present networks where auxiliary flow eradicates the inefficiency of the Wardrop equilibrium (Section 2). However, it turns out that both the *optimal auxiliary flow* of given value and the *minimal amount of an optimal auxiliary flow* are NP-hard to compute (Subsection 3.1 and 3.2). Further, we prove that for auxiliary flow there is no polynomial time approximation with a factor of less than $\frac{4}{3}$. The minimal amount of the optimal auxiliary flow cannot be approximated by any subexponential factor. These results are complemented by proving NP-hardness for adversarial flow (Subsection 3.3).

Related Work. The game theoretic traffic model considered in this paper was introduced by Wardrop [29]. Beckmann et al. [2] observe that such an equilibrium flow is an optimal solution to a related convex program. They give existence and uniqueness results for traffic equilibria (see also [9] and [24]). Dafermos and Sparrow [9] show that the equilibrium state can be computed efficiently under some assumptions on the latency functions.

The inefficiency of Wardrop equilibria is a well-known phenomenon [20], which is exemplified by Braess paradox [3]. Bounding the inefficiency of equilibria, however, has only recently been considered, initiated by Koutsoupias and Papadimitriou [18], and for the Wardrop model by Roughgarden and Tardos [24, 26].

One of the most prominent approaches to eradicate the inefficiency of Wardrop equilibria is taxing. The effectiveness of taxes has been observed by Pigou [20] and generalized by Beckmann et al. [2]. They show that *marginal cost pricing* completely eliminates the inefficiency of selfish routing. Major results for taxes for heterogeneous users can be found in [8], [10], [11] and [14]. Cole et al. [7] consider taxes that minimize the total user disutility (latency plus tax) at equilibrium. They show that for linear latency functions marginal cost pricing does not improve the cost of Wardrop equilibria and prove tight inapproximability results for optimal taxes.

Korilis et al. [16] consider the problem of a Stackelberg leader, who in a first phase can fix the routes for a certain fraction of the demand. In a second phase, selfish users enter the system and route their own flow on top of the leader demand. The objective of the leader is to minimize the resulting cost of the total (both leader and selfish) flow. Roughgarden [22] shows that it is weakly NP-hard to compute the optimal leader strategy even for parallel links with linear latency functions. Kaporis and Spirakis [13] show that for single-commodity networks the minimal fraction of flow needed by the leader to induce optimal cost, can be computed in polynomial time. Sharma and Williamson [27] compute the minimum fraction of users that must be centrally routed to improve the quality of the resulting Wardrop equilibrium. Subsequent papers [28, 15, 4] consider Stackelberg routing in different variants for more general networks.

Roughgarden [25] studies designing networks that exhibit good performance when used selfishly and proves tight inapproximability results.

Other approaches for coping with selfishness are, for example, proposed by Korilis et al. [17] who add capacity to the resources and Cocchi et al. [6] who study the role of various pricing policies in networks with selfish users.

While the fundamental assumptions is that all agents act selfishly, large systems often display forms of altruism or spite. In these cases, some agents' goals is to improve or to harm the global outcome instead of optimizing their personal objective function. Babaioff et al. [1] and Roth [21] study the existence of equilibria for these games, and quantify the impact of malicious players on the equilibrium. Chen and Kempe [5] proved that equilibria exist for any population of selfish, altruistic and spiteful agents.

2 Preliminaries and Initial Results

We first define the classical Wardrop model originally introduced in [29] and then introduce our additional spam flow. We are given a directed graph $G = (V, E)$ with vertex set V, edge set E, a set of commodities $[k] = \{1, \dots, k\}$ specified by source-sink pairs $(s_i, t_i) \in V \times V$, and flow demands $d_i > 0$. The edges are equipped with non-decreasing, continuous latency functions $\ell_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$.

Let $\mathcal{P}_i$ denote the available paths of commodity i, i. e., all paths connecting s_i and t_i, and let $\mathcal{P} = \bigcup_{i \in [k]} \mathcal{P}_i$. A non-negative path flow vector $(f_P)_{P \in \mathcal{P}}$ is *feasible* if it satisfies the flow demands $\sum_{P \in \mathcal{P}_i} f_P = d_i$ for all $i \in [k]$. Throughout this paper, we will consider only feasible path flow vectors. For single commodity networks we drop the index i and normalize the demand to one. A path flow vector $(f_P)_{P \in \mathcal{P}}$ induces an edge flow vector $f = (f_e)_{e \in E}$ with $f_e = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i : e \in P} f_P$. The latency of an edge $e \in E$ is given by $\ell_e(f_e)$ and the latency of a path P is given by the sum of the edge latencies $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$.

The latency cost of a flow is defined as $C(f) = \sum_{P\in\mathcal{P}} \ell_P(f)f_P = \sum_{e\in E} \ell_e(f_e)f_e$. A flow f of minimal latency cost is called *(socially) optimal*.

Additionally to the given selfish flow, we introduce two kinds of spam flow - auxiliary and adversarial flow (δ_e). Let $\delta > 0$ denote the spam flow and its demand. The objective of the spam flow is to minimize/maximize the latency cost of the induced equilibrium of the selfish flow. The routed spam has no intrinsic value and hence does not contribute to the latency cost. Given the routes of the spam flow and the selfish flow, the latency cost equals $C(f, \delta) = \sum_{e\in E} \ell_e(f_e + \delta_e)f_e$. If not specified further, we refer by flow to the selfish flow. Finally, we call the tuple $\Gamma = (G, (s, t), \delta)$ an *instance*.

A flow vector is considered stable when no fraction of the flow can improve its sustained cost by moving unilaterally to another path. Such a stable state is generally known as *Nash equilibrium*. In our model a flow is stable if and only if all used paths within a commodity have the same minimal latency, whereas unused paths may have larger latency. We call such a flow *Wardrop equilibrium*.

Definition 1. *Given an instance Γ and fixed routes for the spam δ, a feasible flow vector f is at* Wardrop equilibrium *if for every commodity $i \in [k]$ and paths $P_1, P_2 \in \mathcal{P}_i$ with $f_{P_1} > 0$ it holds that $\ell_{P_1}(f + \delta) \leq \ell_{P_2}(f + \delta)$.*

Observation 1. *If f is at Wardrop equilibrium then all used paths in commodity i have equal latency $L_i(f, \delta)$ and the latency cost can be expressed as $C(f, \delta) = \sum_{i\in[k]} L_i(f, \delta) \cdot d_i$ ([24, 29]).*

Note that the spam commodity δ is not composed of stabilizing selfish users. Instead, the aim is to allocate this flow in a coordinated way to influence the cost of the Wardrop equilibrium. Our optimization problem is similar to Stackelberg routing [16]. In particular, it can be formulated as a bilevel problem, where in a first phase spam flow is allocated to the routes. In a second phase the selfish flow

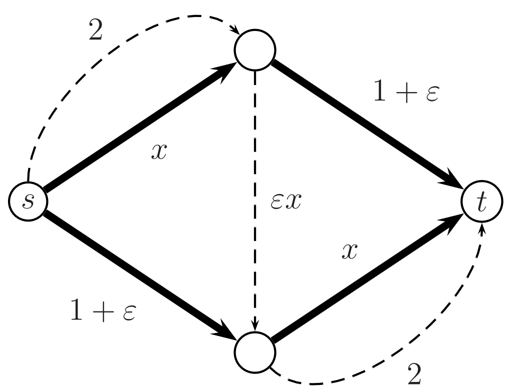


Fig. 1. In absence of spam flow, the selfish flow uses only the zig-zag-path at equilibrium. Routing spam over the dashed edges, the selfish flow splits half-half among the bold edges and reaches the social optimum.

stabilizes at Wardrop equilibrium depending on the allocation in the first phase. The resulting latency of the selfish flow is to be optimized by the allocation of spam flow in the first place.

Let us note two initial observations about auxiliary flow. Figure 1 yields our first observation.

Observation 2. *There are networks in which auxiliary flow eradicates the inefficiency of selfish routing.*

One can easily modify the network in Figure 1, such that even an arbitrary small amount of auxiliary flow does the job.

Observation 3. *Adding auxiliary flow to selfish flow increases the path latency in series-parallel graphs. Since the cost at equilibrium equals the path latency L, auxiliary flow of arbitrary value does not improve the latency cost at equilibrium.*

3 Computational Complexity

In this section, we discuss the computational complexity of problems related to auxiliary and adversarial flow.

In the decision problem OPTIMAL-FLOW we are given a single-source selfish routing instance, an amount of auxiliary flow, and a cost value C. The problem is to decide if there is a routing of the auxiliary flow such that the latency cost of the equilibrium is at most C.

In the decision problem THRESHOLD-FLOW we are given a single-source selfish routing instance and an amount of auxiliary flow δ. The problem is to decide if there is a routing of the auxiliary flow such that the latency cost of the equilibrium is less or equal than the latency cost of the equilibrium induce by any auxiliary flow $\delta' > \delta$.

In the decision problem WORST-FLOW we are given a single-source selfish routing instance, an amount of adversarial flow, and a cost value C. The problem is to decide if there is a routing of the adversarial flow such that the latency cost of the equilibrium is at least C.

3.1 Complexity of OPTIMAL-FLOW

Observation 2 shows that auxiliary flow can improve the cost of Wardrop equilibria. Here, we show that computing the optimal routing for the auxiliary flow is NP-hard.

Theorem 1. OPTIMAL-FLOW *is* NP*-hard.*

Proof. Our proof is based on the proof given in [7] to show that taxing to minimize total disutility is hard. We reduce from the problem 2 DIRECTED DISJOINT PATH (2DDP) which is known to be NP-hard [12]. An instance $I = (G, (s_1, t_1), (s_2, t_2))$ is a directed graph G and two pairs of nodes (s_1, t_1) and

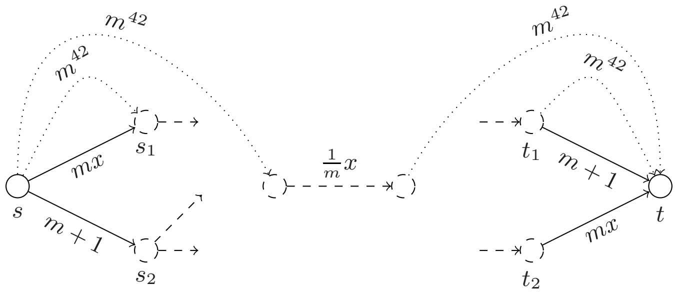


Fig. 2. This figure outlines the construction of G'. The dashed edges are the edges of G and the dotted edges are the edges in P. The edges are labeled with their latency functions.

(s_1, t_2). An instance I belongs to 2DDP, that is $I \in$ 2DDP, if and only if there exist two node disjoint paths in G from s_1 to t_1 and from s_2 to t_2, respectively. Without loss of generality, we assume that there exist paths from s_1 to t_1 and from s_2 to t_2, respectively.

Given an instance $I = (G, (s_1, t_1), (s_2, t_2))$ with $G = (V, E)$ and $|E| = m$, we construct a single commodity selfish routing game $\Gamma = (G', (s, t), \delta)$ with auxiliary flow of $\delta = 3m^2$ that has the following properties: If and only if $I \in$ 2DDP, optimal auxiliary flow yields a Wardrop equilibrium with social cost of less than $C = \frac{3}{2}m + 2$.

We construct $G' = (V', E')$ as follows: $V' = V \cup \{s, t\}$ and $E' = E \cup \{(s, s_1), (s, s_2), (t_1, t)(t_2, t)\} \cup P$ with $P = \{(s, u), (v, t) \mid \text{for all } (u, v) \in E\}$. The latency function of each edge $e \in E$ is $\ell_e(x) = \frac{1}{m}x$, for the edges $e \in \{(s, s_1), (t_2, t)\}$ it is $\ell_e(x) = mx$, for the edges $e \in \{(s, s_2), (t_1, t)\}$ it is $\ell_e(x) = m + 1$, and for all edges $e \in P$ it is $\ell_e(x) = m^{42}$. Note that in equilibrium no selfish flow is assigned to an edge $e \in P$ because latency of m^{42} is much larger than the latency of any s-t-path that does not include an edge $e \in P$.

If $I \in$ 2DDP, there exist two disjoint paths from s_1 to t_1 and from s_2 to t_2, respectively, in G'. Let $D \subseteq E$ be the set of edges of these two paths. An auxiliary flow that assigns, for all $(u, v) \in E \setminus D$, flow of at least $3m$ to each of the edges $(s, u), (v, t) \in P$, and (u, v) essentially forces the selfish flow to use the two disjoint paths only. The latency for flow demand d' on such a path is at least $md' + m + 1$ and at most $md' + m \cdot \frac{1}{m} + m + 1$. Thus, in equilibrium the maximal flow demand on each of the two paths is bounded by $\frac{m+1}{2m+1}$. Therefore, the latency of a path in a resulting Wardrop equilibrium is at most $\frac{3}{2}m + 2$ and the latency cost is at most C.

If $I \notin$ 2DDP, we show that there is no auxiliary flow that induces an equilibrium flow with social cost of less than $2m$. We distinguish several cases by the usage of the four edges incident to s and t. It suffices to show that there is an used path with latency of at least $2m$.

1. If a flow uses a path starting with (s, s_2) and ending with (t_1, t), this path has latency of at least $2m + 2$.

2. If a flow uses only paths starting with (s, s_1) and ending with (t_2, t), it has cost of at least $2m$.
3. If a flow uses only paths starting with (s, s_1) and ending with (t_2, t) or (t_1, t), the latency from s_1 to t must be the same on all paths. Therefore every path has latency of at least $2m + 1$.
4. If a flow uses only paths starting with (s, s_1) or (s, s_2) and ending with (t_2, t), the same argument holds.
5. If a flow uses at least one path starting with (s, s_1) and ending with (t_1, t) and at least one path starting with (s, s_2) and ending with (t_2, t), there exists a vertex v^* that is contained in both paths. All path segments from s to v^* and from v^* to t must have the same latency. Thus, every path has latency of at least $2m + 2$.

Thus, the optimal auxiliary flow induces an equilibrium with social cost less or equal C in Γ if and only if $I \in$ 2DDP. □

Note that the decision in the previous instances is whether the cost of the selfish flow can be reduced to a cost of at most $C = \frac{3}{2}m + 2$. If this is impossible, for every flow the cost is at least $2m$. Now suppose there is a polynomial time approximation algorithm, which computes a $\left(\frac{4}{3} - \epsilon\right)$-approximation for optimizing the cost of selfish flow. Then, such an algorithm could be used to decide 2DDP using the previously outlined set of instances. We therefore get the following corollary. Note that a $\frac{4}{3}$-approximation for linear latencies is trivially obtained by routing no auxiliary flow at all [24].

Corollary 4. *For every $\epsilon > 0$ it is* NP*-hard to approximate* Optimal-Flow *on instances with linear latency functions to a factor of $\frac{4}{3} - \epsilon$.*

In addition, note that in the NP-hardness reduction the auxiliary flow is much larger than the demand of selfish flow. However, we can show that the result even holds, if the auxiliary flow is only a (polynomially small) fraction of the selfish demand.

Theorem 2. Optimal-Flow *is* NP*-hard and even* NP*-hard to approximate to a factor of $\frac{4}{3} - \epsilon$ for every $\epsilon > 0$ on instances with linear latency functions and auxiliary flow $\delta \in \mathcal{O}\left(\frac{d}{poly(m)}\right)$.*

Proof. Again, we reduce from 2DDP. Given an instance I and an ϵ, we construct a selfish routing game Γ as described in the proof of Theorem 1. We use $k = 3m^2 \cdot \lceil \epsilon^{-1} \rceil$ copies $\Gamma_1, \ldots, \Gamma_k$ of this game to create a new game Γ' as follows. We add a source vertex s^* and a target vertex t^*. The vertex s^* is connected to each source vertex s'_i of Γ_i (for all $1 \leq i \leq k$) by an edge (s^*, s_i) with the latency function $\ell_{(s^*, s_i)}(x) = 0$. Likewise, there is an edge with $\ell_{(t'_i, t^*)}(x) = 0$ from each vertex t'_i to t^*. Additionally, for every $i \in \{1, \ldots, k-1\}$, there is an edge from t'_i to s'_{i+1} with $\ell_{(t'_i, s_{i+1})}(x) = k^{42}$. The demand of the selfish flow is $d = k$ and the auxiliary flow is limited to $3m^2$ and $C = d \cdot \frac{3}{2}m + 2$.

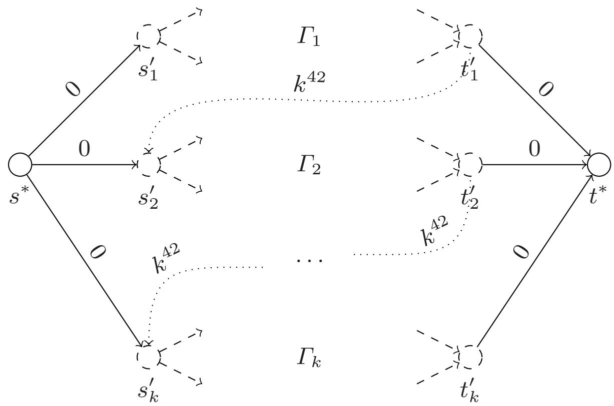


Fig. 3. The network contains $k = 3m^2 \cdot \lceil \epsilon^{-1} \rceil$ copies of the network G' of the proof of Theorem 1. Between s^* and t^* there is a demand of k.

If $I \in$ 2DDP, there is an auxiliary flow that yields an equilibrium flow with social costs of at most $d \cdot (\frac{3}{2}m + 2)$: We assign auxiliary flow of at most $3m^2$ between the vertices s'_i and t'_i in each copy Γ_i as described in the proof of Theorem 1. We assign the same amount of flow to the edges $\{(s^*, s'_1), (t'_1, s'_2), \ldots, (t'_{k-1}, s'_k), (t'_k, t^*)\}$ to obtain a flow of at most $3m^2$ from s^* to t^*. In the resulting Wardrop equilibrium, there is a flow of 1 that is assigned to each copy Γ_i and the edges that connect it to s^* and t^*. Each of these flows has cost of at most $\frac{3}{2}m + 2$. Thus the social cost sum up to at most $d \cdot (\frac{3}{2}m + 2)$.

If $I \notin$ 2DDP, then the latency cost of the selfish flow is more than $d \cdot 2m$. Note, that in equilibrium the selfish flow never chooses an edge that connects two of the copies because it has latency of k^{42} and there is always a s^*-t^*-path with lower latency. Therefore, there is at least one copy Γ_i in which flow of at least 1 is routed from s'_i to t'_i. As shown in the proof of Theorem 1, the latency of the s'_i-t'_i-paths at least $2m$. Since the flow is a Wardrop equilibrium, every path between s'_j and t'_j for every $1 \leq j \leq k$ has latency of at least $2m$. Thus, the latency cost sums up to more than $d \cdot 2m$. □

3.2 Complexity of Threshold-Flow

The previous result showed that it is computationally difficult to compute the best possible auxiliary flow. In this section we show that it is even hard to approximate the minimal amount of auxiliary flow that is needed to achieve the best possible Wardrop equilibrium.

Note that this result strongly contrasts the corresponding result of Kaporis and Spirakis [13] for Stackelberg routing. In Stackelberg routing the minimal fraction of flow needed by the Stackelberg leader to induce optimal cost can be computed in polynomial time for single commodity networks.

Theorem 3. Threshold-Flow *is* NP*-hard.*

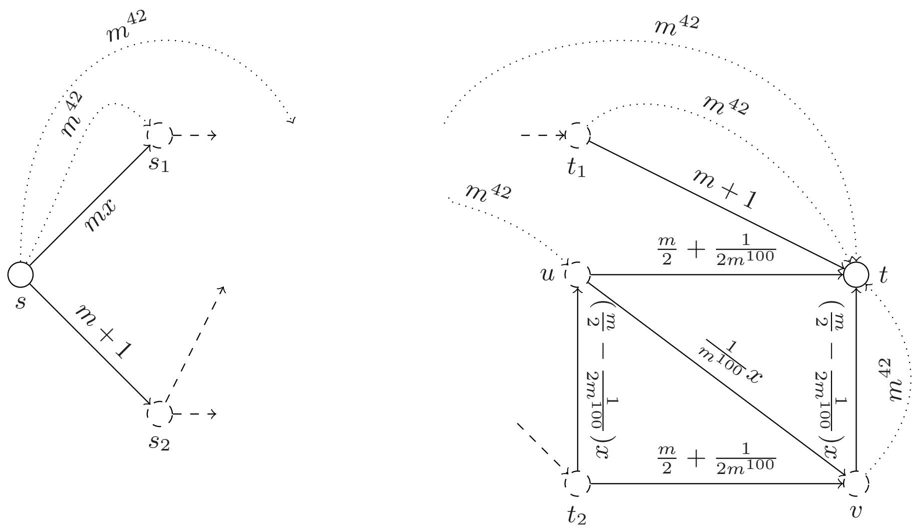


Fig. 4. This figure outlines the modified construction of G' for the proof of Theorem 3

Proof. Again, we reduce from 2 Directed Disjoint Path (2DDP). Given an instance $I = (G, (s_1, t_1), (s_2, t_2))$ with $G = (V, E)$ and $|E| = m$, we construct a single commodity selfish routing game whose optimal auxiliary flow has demand of at most $3m^3$ if and only if $I \in$ 2DDP. Construct $\Gamma = (G', (s, t), \delta)$ as described in the proof of Theorem 1 and modify it as follows. Remove the edge (t_2, t) and replace it with the following gadget. Add the vertices u and v and the edges $(t_2, u), (u, v), (u, t), (t_2, v), (v, t)$. Latency functions are $\ell_e(x) = (\frac{m}{2} - \frac{1}{2m^{100}})x$ for the edges $e \in \{(t_2, u), (v, t)\}$ and $\ell_e(x) = \frac{m}{2} + \frac{1}{2m^{100}}$ for the edges $e \in \{(u, t), (t_2, v)\}$ and $\ell_{(u,v)}(x) = \frac{1}{m^{100}}x$. We add additional edges (s, u) and (v, t) with latency m^{42} and add them to the set P (cf. proof of Theorem 1).

Observe that for routing flow demand $d' \leq \frac{2m^{101}+2}{3m^{101}+1}$ from t_2 to t, it is optimal to leave all selfish flow on the zig-zag path, which generates latency md' and also yields an equilibrium. Observe that the optimum assignment of selfish flow that is achievable by (marginal cost) taxing might split the flow along all three possible paths from t_2 to t. However, the resulting latency of such a flow is larger here, as the auxiliary flow is accounted in the latency of selfish flow. For more flow than d', splitting the flow and assigning $\frac{d'}{2}$ to the edges (t_2, u), (t_2, v), (u, t), and (v, t), yields a better latency. This flow and its improved latency can be achieved using a sufficiently large auxiliary flow along edge (u, v).

If $I \in$ 2DDP, then optimal auxiliary flow of demand of at most $3m^3$ is sufficient to obtain the best possible Wardrop equilibrium. Note that for large m only close to $\frac{1}{2}$ selfish flow is routed through the gadget from t_2 to t. Therefore, it is not necessary to route auxiliary flow over the edge (u, v).

If $I \notin$ 2DDP, then optimal auxiliary flow yields a Wardrop equilibrium in which the whole selfish demand is routed from s via s_1 and t_2 to t. The optimal

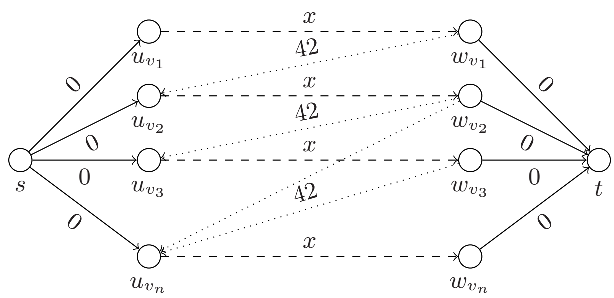


Fig. 5. This figure depicts the corresponding graph G' for an instance G for the problem HAMILTON. The dashed edges correspond to vertices in G and the dotted edges correspond to edges in G'.

auxiliary flow must block edge (u, v). Even to motivate one selfish player to use (u, t), it needs to route demand of more than $\frac{m^{101}}{2} - \frac{1}{2}$ over the edge (u, v). □

Note that one can easily replace the term m^{100} in the latency functions of our gadget with any arbitrarily large constant that can be represented by a polynomial number of bits in the input size. In particular, assuming that the numbers in our instance are represented in binary coding, it can be replaced by 2^m. Then, for $m \geq 2$, a $\frac{2^m}{6m^3}$-approximation algorithm of THRESHOLD-FLOW in the above instances can decide 2DDP. Thus, we have the following corollary.

Corollary 5. *For any constant $\epsilon > 0$, it is* NP*-hard to approximate* THRESHOLD-*Flow by a factor of* $2^{m(1-\epsilon)}$.

3.3 Complexity of WORST-FLOW

We have seen that the optimal auxiliary flow is NP-hard to compute. We now turn to the computational complexity of computing the optimal adversarial flow.

Theorem 4. WORST-FLOW *is* NP*-hard.*

Proof. We reduce from the NP-hard problem HAMILTON. A graph $G \in$ HAMILTON if and only if G contains a Hamiltonian path. Given a directed graph $G = (V, E)$ with $|V| = n$ and $|E| = m$ and two vertices $x, y \in V$, we construct a selfish routing game $\Gamma = (G', (s, t), \delta)$ that has the property that the cost maximizing adversarial flow induces social cost of at least $C = \frac{1}{n} + \delta$ if and only if $G \in$ HAMILTON. We construct $G' = (V', E')$ as follows: For every node v in G there is a a pair of nodes u_v, w_v in G' and, additionally we have a source and a sink node s and t. That is $V = \{s, t\} \cup \{u_v, w_v \mid \forall v \in V\}$.

There are edges from s to all u nodes, from each node u_v to w_v and from all w nodes to t. The selfish flow will use only these edges. Additionally, we have edges (with high latency) that connect a node w_v with a node $u_{v'}$ if there is an edge from v to v' in the graph G for $v \in V - \{x\}$. To summarize $E' = S' \cup U' \cup W'$ with $S' = \{(u_v, w_v) \mid \forall v \in V\}$, $U' = \{(s, u_v), (w_v, t) \mid \forall v \in V\}$, and $W' = \{w_v, u_{v'} \mid \forall (v, v') \in E \text{ and } v' \in V - \{x\}\}$. For all edges $e \in S'$ we set $\ell_e(x) = x$, for all edges $e \in U'$ we set $\ell_e(x) = 0$, and for all edges $e \in W'$ we set $\ell_e(x) = 42$. Note that the selfish flow never uses edges $e \in W'$ and therefore, assigns flow to the n paths s, u_v, w_v, s (for all $v \in V$). Without adversarial flow, the equilibrium flow is equally distributed among these paths and the social costs are $n\frac{1}{n^2} = \frac{1}{n}$.

Assume $G \in$ HAMILTON and $x = v_{i_1}, \ldots, v_{i_n} = y$ is a Hamiltonian path in G. Then it is possible to assign adversarial flow of δ to all edges $e \in S'$ by choosing the path $s, u_{v_{i_1}}, w_{v_{i_1}}, u_{v_{i_2}}, w_{v_{i_2}}, \ldots, u_{v_{i_n}}, w_{v_{i_n}},, t$. Note, that the edges between the w and u vertices exist by construction. Because all non constant edges carry the maximal amount of adversarial flow and this flow maximizes the social costs which are $m(\frac{1}{n} + \delta) \cdot \frac{1}{n} = \frac{1}{n} + \delta$.

Consider a graph $G \notin$ HAMILTON. Then there is no path in G' from s to t that visits all vertices $e \in U'$. Therefore, there is at least one edge with adversarial flow less than δ. Thus, the latency cost of the equilibrium flow is strictly less than $\frac{1}{n} + \delta$. □

4 Conclusions

We have initiated the study of spam flow in non-atomic routing games. We considered the computational complexity of several problems related to auxiliary and adversarial flow. Both, OPTIMAL-FLOW and WORST-FLOW turned out to be NP-hard. Moreover, OPTIMAL-FLOW and THRESHOLD-FLOW are hard to approximate, which strongly contrasts the results for the analogous problem of the "Price of Optimum" in Stackelberg routing [13]. Further research on algorithms and corresponding complexity issues regarding spam that improves or deteriorates latency cost may well be worthwhile.

References

[1] Babaioff, M., Kleinberg, R., Papadimitriou, C.: Congestion games with malicious players. In: Proc. 8th EC, pp. 103–112 (2007)

[2] Beckmann, M., McGuire, C.B., Winston, C.B.: Studies in the Economics of Transportation. Yale University Press, New Haven (1956)

[3] Braess, D.: Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung 12, 258–268 (1968)

[4] Bonifaci, V., Harks, T., Schäfer, G.: Stackelberg routing in arbitrary networks. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 239–250. Springer, Heidelberg (2008)

[5] Chen, P., Kempe, D.: Altruism, selfishness, and spite in traffic routing. In: Proc. 9th EC, pp. 140–149 (2008)

[6] Cocchi, R., Shenker, S., Estrin, D., Zhang, L.: Pricing in computer networks: motivation, formulation, and example. IEEE/ACM Transactions on Networking 1(6), 614–627 (1993)

[7] Cole, R., Dodis, Y., Roughgarden, T.: How much can taxes help selfish routing? In: Proc. 4th EC, pp. 98–107 (2003)
[8] Cole, R., Dodis, Y., Roughgarden, T.: Pricing edges for heterogeneous selfish users. In: Proc. 35th STOC, pp. 521–530 (2003)
[9] Dafermos, S., Sparrow, F.T.: The Traffic Assignment Problem for a General Network. J. Research National Bureau of Standards, Series B 73(2), 91–118 (1969)
[10] Fleischer, L.: Linear tolls suffice: New bounds and algorithms for tolls in single source networks. Theoretical Computer Science 348, 217–225 (2005)
[11] Fleischer, L., Jain, K., Mahdian, M.: Taxes for heterogeneous selfish users in a multicommodity network. In: Proc. 45th FOCS, pp. 277–285 (2004)
[12] Fortune, S., Hopcroft, J.E., Wyllie, J.: The Directed Subgraph Homeomorphism Problem. Theoretical Computer Science 10, 111–121 (1980)
[13] Kaporis, A.C., Spirakis, P.G.: The price of optimum in stackelberg games on arbitrary single commodity networks and latency function. In: Proc. 18th SPAA (2006)
[14] Karakostas, G., Kolliopoulos, S.G.: Edge pricing of multicommodity networks for heterogeneous selfish users. In: Proc. 45th FOCS, pp. 268–276 (2004)
[15] Karakostas, G., Kolliopoulos, S.G.: Stackelberg strategies for selfish routing in general multicommodity networks. Algorithmica (to appear) (2008)
[16] Korilis, Y.A., Lazar, A.A., Orda, A.: Achieving network optima using Stackelberg routing strategies. IEEE/ACM Transactions on Networking 5(1), 161–173 (1997)
[17] Korilis, Y.A., Lazar, A.A., Orda, A.: Capacity allocation under noncooperative routing. IEEE Transactions on Automatic Control 42, 309–325 (1997)
[18] Koutsoupias, E., Papadimitriou, C.: Worst-Case Equilibria. In: Proc. 16th STACS, pp. 404–413 (1999)
[19] Kumar, V.S.A., Marathe, A.: Improved results for Stackelberg scheduling strategies. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 776–787. Springer, Heidelberg (2002)
[20] Pigou, A.C.: The Economics of Welfare. Macmillan, Basingstoke (1920)
[21] Roth, A.: The Price of Malice in Linear Congestion Games. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 118–125. Springer, Heidelberg (2008)
[22] Roughgarden, T.: Stackelberg scheduling strategies. In: Proc. 33rd STOC, pp. 104–113 (2001)
[23] Roughgarden, T.: The Price of Anarchy is Independent of the Network Topology. In: Proc. 34th STOC, pp. 428–437 (2002)
[24] Roughgarden, T., Tardos, É.: How Bad is Selfish Routing? J. ACM 49(2), 236–259 (2002)
[25] Roughgarden, T.: On the Severity of Braess's Paradox: Designing Networks for Selfish Users is Hard. JCSS 72(5), 922–953 (2006)
[26] Roughgarden, T.: Selfish Routing and the Price of Anarchy. MIT Press, Cambridge (2005)
[27] Sharma, Y., Williamson, D.P.: Stackelberg thresholds in network routing games or the value of altruism. In: Proc. 8th EC, pp. 93–102 (2007)
[28] Swamy, C.: The effectiveness of Stackelberg strategies and tolls for network congestion games. In: Proc. 18th SODA, pp. 1133–1142 (2007)
[29] Wardrop, J.G.: Some Theoretical Aspects of Road Traffic Research. In: Proc. of the Institute of Civil Engineers, Pt. II, pp. 325–378 (1952)

On Profit-Maximizing Pricing for the Highway and Tollbooth Problems

Khaled Elbassioni[1], Rajiv Raman[1], Saurabh Ray[1], and René Sitters[2]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
{elbassio,rraman,saurabh}@mpi-inf.mpg.de
[2] Department of Econometrics and Operations Research, VU University, Amsterdam, the Netherlands
rsitters@feweb.vu.nl

Abstract. In the *tollbooth problem on trees*, we are given a tree $\mathbf{T} = (V, E)$ with n edges, and a set of m customers, each of whom is interested in purchasing a path on the graph. Each customer has a fixed budget, and the objective is to price the edges of $\mathbf{T}$ such that the total revenue made by selling the paths to the customers that can afford them is maximized. An important special case of this problem, known as the *highway problem*, is when $\mathbf{T}$ is restricted to be a line. For the tollbooth problem, we present an $O(\log n)$-approximation, improving on the current best $O(\log m)$-approximation. We also study a special case of the tollbooth problem, when all the paths that customers are interested in purchasing go towards a fixed root of $\mathbf{T}$. In this case, we present an algorithm that returns a $(1 - \epsilon)$-approximation, for any $\epsilon > 0$, and runs in quasi-polynomial time. On the other hand, we rule out the existence of an FPTAS by showing that even for the line case, the problem is strongly NP-hard. Finally, we show that in the *discount model*, when we allow some items to be priced below zero to improve the overall profit, the problem becomes even APX-hard.

1 Introduction

Consider the problem of pricing the bandwidth along the links of a network such that the revenue obtained from customers interested in buying bandwidth along certain paths in the network is maximized. Suppose that each customer declares a set of paths she is interested in buying, and a maximum amount she is is willing to pay for each path. The network service provider's objective is to assign single prices to the links such that the total revenue from customers who can afford to purchase their paths is maximized. Recently, numerous papers have appeared on the computational complexity of such pricing problems [1,6,5,7,8,9,10,11,12,15,13,16,17,18].

A special case of this problem, where each customer is interested in purchasing only a single path (*single-minded*), and where there is no upper bound on the number of customers purchasing each link (*unlimited supply*) was studied by Guruswami et al. [16], under the name of *tollbooth problem*. The authors of [16]

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 275–286, 2009.

showed that the problem is already APX-hard when the network is restricted to be a tree, and also presented a polynomial time algorithm for the case when all paths start at a certain root of the tree. In [16], the authors also studied the *highway problem*, a further restriction where the tree is a path, and gave polynomial time algorithms when either the budgets are bounded and integral, or all paths have a bounded length.

In this paper, we continue the study of these problems. For the tollbooth problem, a uniform pricing gives an approximation factor of $O(\log n + \log m)$, where n and m are respectively the number of edges of the tree and the number of customers. This result applies in fact for general sets [16], and not necessarily paths of a network, and even in the non single-minded case [4]. Very recently, and more generally, Cheung and Swamy [10] gave an algorithm that, given any LP-based α-approximation algorithm for maximizing the social welfare under limited supply, returns a solution with profit within a factor of $\alpha \log u_{max}$ of the maximum, where u_{max} is the maximum supply of an item. In particular, this gives an $O(\log m)$-approximation for the tollbooth problem on trees. In this paper, we give an $O(\log n)$-approximation which is an improvement over the $O(\log m)$ since $n \leq 3m$ can be always assumed (see Section 2). While the problem is APX-hard even in the very simple case of a star [16], we show that if all the paths are going towards (but not necessarily starting at) a certain root, then a $(1 - \epsilon)$-approximation can be obtained in quasi-polynomial time. This result extends a recently developed quasi-PTAS [12] for the highway problem, and uses essentially the same technique. However, there is a number of technical issues that have to be resolved for this technique to work on trees; most notably is the use of the Separator Theorem for trees, and the modification of the price-guessing strategy to allow only for *one-sided* guesses.

The existence of a quasi-PTAS for the highway problem indicates that a PTAS or even an FPTAS is still a possibility, since the problem was only known to be weakly NP-hard [8]. In the last section of this paper, we show that the highway problem is indeed strongly NP-hard and hence admits no FPTAS unless P=NP.

Balcan et al. [3] considered a model in which some items can be priced below zero (in the form of a discount) so that the overall profit is maximized. They gave a 4-approximation for the uniform budgets case, and a quasi-PTAS for a special case in which there is an optimal pricing that has only a bounded number of negatively priced items. Here we show that the existence of a quasi-PTAS in the general case is highly unlikely, by showing that the problem is APX-hard.

In the next section, we give a formal definition of the problem. In Section 3, we give an $O(\log n)$ approximation for trees and in Section 4 we give a quasi-PTAS for the case of uncrossing paths. We describe our hardness results in Section 5. We conclude in Section 6. Due to lack of space, most proofs have been omitted from this extended abstract.

2 The Tollbooth Problem on Trees

2.1 Notation

Let $\mathbf{T} = (V, E)$ be a tree. We assume that we are given a (multi)set of paths $\mathcal{I} = \{I_1, \ldots, I_m\}$, defined on the set of edges E, where $I_j = [s_j, t_j] \subseteq E$ is the path connecting s_j and t_j in $\mathbf{T}$. For $I_j \in \mathcal{I}$, we denote by $B(I_j) \in \mathbb{R}_+$ the *budget* of path I_j, i.e., the maximum amount of money customer j is willing to pay for purchasing path I_j. In the *tollbooth problem*, denoted henceforth by TB, the objective is to assign a price $p(e) \in \mathbb{R}_+$ for each edge $e \in E$, and to find a subset $\mathcal{J} \subseteq \mathcal{I}$, so as to maximize $\sum_{I \in \mathcal{J}} p(I)$ subject to the budget constraints

$$p(I) \leq B(I), \quad \text{for all } I \in \mathcal{J}, \tag{1}$$

where, for $I \in \mathcal{I}$, $p(I) = \sum_{e \in I} p(e)$.

For a node $w \in V$, let $\mathcal{I}[w] \subseteq \mathcal{I}$ be the set of paths that pass through w. In section 4, we will assume that the tree is rooted at some node $\mathbf{r} \in V$. The depth of $\mathbf{T}$, denoted $d(\mathbf{T})$, is the length of the longest path from the root $\mathbf{r}$ to a leaf. For a node $w \in V$, we denote by $\mathbf{T}(w)$, the subtree of $\mathbf{T}$ rooted at w (excluding the path from the parent of w to $\mathbf{r}$), and for a subtree $\mathbf{T}'$ of $\mathbf{T}$ we denote by $V(\mathbf{T}'), E(\mathbf{T}')$ and $\mathcal{I}(\mathbf{T}')$ the vertex set, edge set, and set of intervals contained completely in $\mathbf{T}'$, respectively.

2.2 Preliminaries

In the following sections, we denote by $p^* : E \mapsto \mathbb{R}_+$ an optimal set of prices, and by OPT $\subseteq \mathcal{I}$ the set of intervals purchased in this optimum solution. For a subset of intervals $\mathcal{I}' \subseteq \mathcal{I}$, and a price function $p : E \mapsto \mathbb{R}_+$, we denote by $p(\mathcal{I}') = \sum_{I \in \mathcal{I}'} p(I)$ the total price of intervals in $\mathcal{I}'$.

It easy to see that $n \leq 3m$ may be assumed without loss of generality. Indeed, if we root the tree at some vertex $\mathbf{r}$, then for every vertex $v \in V$, we may assume that there is either an interval $I \in \mathcal{I}$ beginning at v or an interval $I \in \mathcal{I}$ that passes through two different children of v; otherwise, every interval through v must contain its parent u (unless $v = \mathbf{r}$ in which case all edges incident to $\mathbf{r}$ can be contracted), and hence we can contract the edge $e = \{u, v\}$ and increase by $p^*(e)$ the prices of each the edges $\{v, v'\}$ for each child v' of v.

Let $\epsilon > 0$ be a given constant.

Proposition 1 ([12]). *Let p^* be an optimal solution for a given instance of* TB, *and $\epsilon > 0$ be a given constant. Then there exists a price function $\tilde{p} : E \mapsto \mathbb{R}_+$ for which*

(i) $\tilde{p}(e) \in \{0, 1, \ldots, P\}$, for every $e \in E$, where $P = nm/\epsilon$,
(ii) $\tilde{p}(I) \leq \frac{B(I)}{1+\epsilon}$, for every $I \in$ OPT, and
(iii) $\tilde{p}(\text{OPT}) \geq (1 - 2\epsilon)p^*(\text{OPT})$.

We shall call the set of prices $\tilde{p}$ satisfying the conditions of Proposition 1, ϵ-*optimal prices*.

We will make use of the following well-known separator result for trees.

Proposition 2. *Let $T = (V, E)$ be a tree. Then there exists a node v (called separator node) with the following property: Let $s_1, \ldots, s_r$ be the sizes of the components obtained by deleting v from $\mathbf{T}$, then there is a subset $S \subseteq [r] \stackrel{\text{def}}{=} \{1, \ldots, r\}$ such that*

$$\lfloor \frac{n}{3} \rfloor \leq \sum_{i \in S} s_i \leq \lceil \frac{2n}{3} \rceil. \tag{2}$$

Such a separator can be found in linear time.

This gives a recursive partitioning of $\mathbf{T}$ in the following standard way: Let v_0 be a separator vertex in $\mathbf{T}$ and $\mathbf{T}_1, \ldots, \mathbf{T}_r$ be the components of $\mathbf{T} - v_0$. Recursively, find separator vertices $v_1, \ldots v_r$ in $\mathbf{T}_1, \ldots, \mathbf{T}_r$. We say that node v_0 has level(v_0) = 1, nodes $v_1, \ldots, v_r$ have level 2, and in general if node v is a separator vertex in the subtree $\mathbf{T}'$ obtained by deleting one-higher level separator vertex v' then level(v) = level(v') + 1. By (2), the maximum number of levels k in this decomposition is at most $\log_{3/2} n$. We shall denote by $\mathcal{N}(\mathbf{T})$ the set of separator nodes used in the full decomposition of $\mathbf{T}$.

3 An $O(\log n)$ Approximation for the Tollbooth Problem on Trees

In this section, we prove the following theorem.

Theorem 1. *There is a deterministic $O(\log n)$-approximation algorithm for* Tb.

The proof goes along the same lines used in [2] to obtain an $O(\log n)$-approximation for the highway problem. The algorithm consists of 3 main steps: Partitioning, "randomized cut", and then dynamic programming. We can then derandomize it to obtain a deterministic algorithm. We give the details below.

We say that the given set of paths $\mathcal{I}$ is *rooted*, if all the paths in $\mathcal{I}$ start at some node $\mathbf{r}$, called the root of $\mathbf{T}$. We will also make use of the following theorem.

Theorem 2 ([16]). *The tollbooth problem on rooted paths can be solved in polynomial time using dynamic programming.*

For $i = 1, \ldots, k$, let

$$\mathcal{I}(i) = \{I \in \mathcal{I} : \ i \text{ is the smallest level of a separator } v \in \mathcal{N}(\mathbf{T}) \text{ contained in } I\}.$$

Then $\mathcal{I} = \cup_{i \in [k]} \mathcal{I}(i)$ and $I \cap J = \emptyset$ for all $I, J \in \mathcal{I}(i)$ that contain distinct separators at level i. Let (Opt, p^*) be an optimal solution. Then, $p^*(\text{Opt}) = \sum_{i=1}^{k} p^*(\text{Opt} \cap \mathcal{I}(i))$. Thus if we solve k independent problems on each of the sets $\mathcal{I}(i)$, $i = 1, \ldots, k$, and take the solution with maximum revenue, we get a solution of value at least $p^*(\text{Opt})/k$. Thus it remains to show the following result.

Theorem 3. *Let v be a node of $\mathbf{T}$, and suppose that all the paths in $\mathcal{I}$ go through v. Then a solution $(\mathcal{J}, p)$ of expected value $p(\mathcal{J}) \geq p^*(\text{Opt})/8$ can be found in polynomial time.*

Proof. Let $v_1, \ldots, v_r$ be the nodes adjacent to v. Note that each path $I \in \mathcal{I}$ can be divided into two sub-paths starting at v; we denote them by I_1 and I_2. We use the following procedure.

1. Let $X \subseteq \{v_1, \ldots, v_r\}$ be a subset obtained by picking each v_i randomly and independently with probability 1/2.
2. Let $\mathcal{I}' = \{I_j : \; I \in \mathcal{I}, \; j \in \{1, 2\}, \; I_j$ contains exactly one vertex of $X\}$.
3. Use dynamic programming (cf. Theorem 2) to get an optimal solution $(\mathcal{J}, p)$ on the instance defined by $\mathcal{I}'$ and the tree $\mathbf{T}'$ with root v and sub-trees rooted at the children in X.
4. Extend p with zeros on all the other arcs not in $\mathbf{T}'$, and return $(\mathcal{J}, p)$.

Let (Opt, p^*) be an optimal solution. We now argue that the solution returned by this algorithm has expected revenue of $p^*(\text{Opt})/8$. Clearly, for every $I \in \mathcal{I}$, either $p^*(I_1) \geq p^*(I)/2$ or $p^*(I_2) \geq p^*(I)/2$; let us call this more profitable part by I_*. Then $\sum_{I \in \text{Opt}} p^*(I_*) \geq p^*(\text{Opt})/2$. Let $\text{Opt}' = \{I_* : \; I \in \text{Opt}, \; I$ contains exactly one vertex of X and this vertex lies on $I_*\}$. Note that with probability exactly 1/4 each $I \in \text{Opt}$ has I_* belonging to Opt'. In particular,

$$\mathbb{E}[p^*(\text{Opt}')] = \sum_{I \in \text{Opt}} \mathbb{E}[p^*(I_*)] = \frac{1}{4} \sum_{I \in \text{Opt}} p^*(I_*) \geq \frac{1}{8} p^*(\text{Opt}).$$

Since what our procedure returns is at least as profitable as this quantity, the theorem follows. □

The randomized algorithm above can be derandomized using the method of *pairwise independence* [19,20,2].

4 Uncrossing Paths

Here we assume that the tree is rooted at some node $\mathbf{r} \in V$, and that paths in $\mathcal{I}$ have the following *uncrossing* property: If $I = [s, t] \in \mathcal{I}$ then t lies on the path $[s, \mathbf{r}]$. This property implies that once paths in $\mathcal{I}$ meet they cannot diverge.

In the course of the solution, we shall consider the following generalized version of the problem: Given intervals as above, and also a function $h : \mathcal{I} \times \mathbb{R}_+^n \mapsto \mathbb{R}_+$, find $\mathcal{J} \subseteq \mathcal{I}$ and a pricing $p : E \mapsto \mathbb{R}_+$, satisfying (1) and maximizing $\sum_{I \in \mathcal{J}} h(I, p)$.

Given a price function $p : E \mapsto \mathbb{R}_+$ and a node $w \in V$, the *accumulative price* at any node u *on the path* $[w, \mathbf{r}]$ with respect to w is defined as $p([w, u])$. Obviously, this monotonically increases as u moves towards the root. In this section we prove the following theorem.

Theorem 4. *There is a quasi-polynomial time approximation scheme for the tollbooth problem with uncrossing paths.*

In the following, we fix $K = \lceil \log(nP)/\log(1+\epsilon) \rceil$, where $P = \frac{nm}{\epsilon}$ (c.f. Proposition 1).

Definition 1. (ϵ-Relative pricings) *Let $w \in V$ be a given node of $\mathbf{T}$, and $0 \leq k \leq K$ and $0 \leq k' \leq 2\log_{3/2} n$ be given integers. We call any selection of k nodes $u_1, \ldots, u_k \in V$, k indices $-\infty \leq i_1 < \cdots < i_k \leq K$, and k' values $p_1, \ldots, p_{k'} \in \{0, 1, \ldots, nP\}$, such that $w, u_1, u_2, \ldots, u_k, \mathbf{r}$ lie on the path $[w, \mathbf{r}]$ in that order, an* ϵ-relative pricing *w.r.t. w, and denote it by $(w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$.*

The total number of possible ϵ-relative pricings with respect to a given $w \in V$ is at most

$$L = (d(T)K)^K (nP + 1)^{2\log_{3/2} n}, \tag{3}$$

which is $m^{\text{polylog}(m)}$ for every fixed $\epsilon > 0$.

Definition 2. (Consistent pricings) *Let $R = (w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$ be an ϵ-relative pricing w.r.t. node $w \in V$, $\mathcal{L} = \{s_1, \ldots, s_{k'}\}$ be the set of separators from $\mathcal{N}(\mathbf{T})$ on the path from $(w, \mathbf{r}]$, and $p : E \mapsto \mathbb{R}_+$ be a pricing of E. We say that R is* ϵ-consistent *with p and $\mathcal{L}$ if*

(C1) for $j = 1, \ldots, k-1$, $(1+\epsilon)^{i_j} \leq p([w, u]) \leq (1+\epsilon)^{i_j+1}$ if u lies in the interval $[u_j, u_{j+1})$ (excluding u_{j+1}),
(C2) for $j = 1, \ldots, k'$, $p([w, s_j]) = p_j$.

Lemma 1. *Let $\tilde{p} : E \mapsto \mathbb{R}_+$ be an ϵ-optimal pricing for a given instance of* Tb*, $w \in V$ be an arbitrary node, and $\mathcal{L} = \{s_1, \ldots, s_{k'}\}$ be the set of separators in $\mathcal{N}(\mathbf{T})$ on the path from $[w, \mathbf{r}]$. Then there exists an ϵ-relative pricing R w.r.t. w, that is ϵ-consistent with $\tilde{p}$ and $\mathcal{L}$.*

With every ϵ-relative pricing R, we can associate a system of linear inequalities, denoted by $S(R)$, on a set of E variables $\{p(e) : e \in E\}$, consisting of the constraints $(C1)$ and $(C2)$, together with the non-negativity constraints $p(e) \geq 0$. The feasible set for this system gives the set of all possible pricings with which R is ϵ-consistent. For two systems of inequalities S_1, S_2, we denote by $S_1 \wedge S_2$ the system obtained by combining their inequalities.

Let $R = (w, k, k', u_1, \ldots, u_k, i_1, \ldots, i_k, p_1, \ldots, p_{k'})$ be an ϵ-relative pricing w.r.t. a node $w \in V$. Given an interval $I \in \mathcal{I}[w]$, we associate a value $v(I, R)$ to I, defined with respect to R as follows: Let $j(I)$ be the largest index such that $u_{i_{j(I)}}$ is contained in I. Then, define $v(I, R) = (1+\epsilon)^{j(I)}$. For a subset of intervals $\mathcal{I}' \subseteq \mathcal{I}$, we define, as usual, $v(\mathcal{I}', R) = \sum_{I \in \mathcal{I}'} v(I, R)$. It follows that for any ϵ-relative pricing R w.r.t. a node $w \in V$, any $p : E \mapsto \mathbb{R}_+$ with which R is consistent, and any $I = [s, t] \in \mathcal{I}[w]$, we have

$$v(I, R) \leq p([w, t]) \leq (1+\epsilon) v(I, R). \tag{4}$$

Decomposition into Two Subproblems. Let $w \in \mathcal{N}(\mathbf{T})$ be a separator node. Then $\mathbf{T}$ can be decomposed into two subtrees $\mathbf{T}_L = (V_L, E_L)$ and $\mathbf{T}_R = (V_R, E_R)$, such that the root $\mathbf{r} \in V_R$ and $w \in V_L \cap V_R$ is the root of $\mathbf{T}_L$. We define two Tb instances $(\mathbf{T}_L, \mathcal{I}_L)$ and $(\mathbf{T}_R, \mathcal{I}_R)$ where:

$$\mathcal{I}_0 = \{[s, t] \in \mathcal{I}[w] \ : \ s \in V_L \text{ and } t \in V_R\},$$
$$\mathcal{I}_L = \{[s, t] \in \mathcal{I} \ : \ s, t \in V_L\} \cup \{[s, w] \ : \ [s, t] \in \mathcal{I}_0\},$$
$$\mathcal{I}_R = \{[s, t] \in \mathcal{I} \ : \ s, t \in V_R\}.$$

In other words, the intervals passing through w, crossing from $\mathbf{T}_L$ to $\mathbf{T}_R$ are truncated in $\mathbf{T}_L$ while all other intervals remain the same[1]. Note that from the choice of w, we have $\max\{|V(\mathbf{T}_L)|, |V(\mathbf{T}_R)|\} \leq \frac{2n}{3} + 1$, and both instances $(\mathbf{T}_L, \mathcal{I}_L)$ and $(\mathbf{T}_R, \mathcal{I}_R)$ are of the uncrossing type, with roots w and $\mathbf{r}$, respectively.

The algorithm is given as Algorithm 1 below. It is initially called with an empty $\mathcal{S}$, and with $h(I) = 0$ for all $I \in \mathcal{I}$. The procedure iterates over all ϵ-relative pricings R, consistent with $\mathcal{S}$, w.r.t. the middle edge e^*, then recurses on the subsets of intervals to the left and right of e^*. Intervals crossing from $\mathbf{T}_L$ to $\mathbf{T}_R$ will be truncated and their values will be charged to $\mathbf{T}_L$; hence the corresponding budgets are reduced, and the corresponding h-values are increased.

Algorithm 1. TB$(\mathbf{T}, \mathcal{I}, \mathbf{r}, B, h, \mathcal{S})$

Require: An uncrossing TB instance $(\mathbf{T} = (V, E), \mathcal{I})$ with root $\mathbf{r}$, budgets and values $B : \mathcal{I} \to \mathbb{R}_+$ and $h : \mathcal{I} \times \mathbb{R}_+^n \to \mathbb{R}_+$, and a feasible system of inequalities $\mathcal{S}$
Ensure: A pricing $p : E \to \mathbb{R}_+$ and a subset $\mathcal{J} \subseteq \mathcal{I}$
1: **if** $|\mathcal{I}| = 0$ **then**
2: $\quad \mathcal{S}' \leftarrow$REDUCE$(\mathcal{S}, E)$
3: $\quad$ return $(p, \emptyset)$, where p is any feasible solution of $\mathcal{S}'$
4: **end if**
5: **if** $d(\mathbf{T}) = 1$ **then**
6: $\quad$ **for** edge e of $\mathbf{T}$ **do**
7: $\quad\quad \mathcal{S}' \leftarrow$REDUCE$(\mathcal{S}, \{e\})$
8: $\quad\quad p(e) \leftarrow \text{argmax}\{\sum_{I \in \mathcal{I}:\ p' \leq B(I)} (h(I) + p') \ : \ p' \text{ satisfies } \mathcal{S}'\}$
9: $\quad\quad \mathcal{J}(e) \leftarrow \{I \in \mathcal{I} : B(I) \geq p(e)\}$
10: $\quad$ **end for**
11: $\quad$ Return $((p(e) : e \in E), \bigcup_{e \in E} \mathcal{J}(e))$
12: **end if**
13: Let w be a separator node of $\mathbf{T}$ and $\mathbf{T}_L, \mathbf{T}_R, \mathcal{I}_0, \mathcal{I}_L, \mathcal{I}_R$ be as defined above
14: **for** every ϵ-relative pricing R w.r.t. w for which $S \wedge S(R)$ is feasible **do**
15: $\quad$ **for** $I \in \mathcal{I}_0$ **do**
16: $\quad\quad B(I) \leftarrow B(I) - (1 + \epsilon)v(I, R)$
17: $\quad\quad h(I) \leftarrow h(I) + v(I, R)$
18: $\quad$ **end for**
19: $\quad (p_1, \mathcal{J}_1) \leftarrow$ TB$(\mathbf{T}_L, \mathcal{I}_L, w, B, h, \mathcal{S})$
20: $\quad (p_2, \mathcal{J}_2) \leftarrow$ TB$(\mathbf{T}_R, \mathcal{I}_R, \mathbf{r}, B, h, \mathcal{S} \wedge S(R))$
21: $\quad$ Let p be the pricing defined by $p(e) = p_1(e)$ if $e \in E_L$ and $p(e) = p_1(e)$ if $e \in E_R$
22: $\quad \mathcal{J} \leftarrow \mathcal{J}_1 \cup \mathcal{J}_2$
23: $\quad$ record $(p, \mathcal{J})$
24: **end for**
25: Return the recorded solution with largest $p(\mathcal{J}) + h(\mathcal{J})$ value

Solving the Base Case. At the lowest level of recursion (either line 3 or 8), we have to solve a linear program defined by the system $\mathcal{S}$. Note that the system may

[1] Throughout, we will make the implicit assumption that each interval has an "identity"; so, for instance, $\mathcal{I}_L \cap \mathcal{I}_0$ will be used to denote the set $\{I \in \mathcal{I}_0 :\ I = [s,t]$ and $[s,w] \in \mathcal{I}_L\}$.

contain constraints on variables outside the current set of edges E of the current tree $\mathbf{T}$ (resulting from previous nodes of the recursion tree). However, we can reduce this LP to one that involves only variables in E. Indeed, any constraint that involves a variable not in E, has the form $L \leq p([w,u]) \leq U$, where $u \in V(\mathbf{T})$, and $w \notin V(\mathbf{T})$ is a separator node such that there is another separator node $w' \in V(\mathbf{T})$ on the path from w to u. Then when w' was considered in the recursion, a constraint of the form $p([w,w')] = q$, for some value q, was appended to $\mathcal{S}$ (recall $(C2)$ in the definition of consistent pricings). Now, we can replace the first constraint by the equivalent constraint $L - q \leq p([w',u]) \leq U - q$, which only involves variables from E. This is exactly what procedure REDUCE$(\mathcal{S},\cdot)$ does in lines 2 and 7.

When the procedure returns, we get a pricing $p : E \mapsto \mathbb{R}_+$ and a set of intervals $\mathcal{J} \subseteq \mathcal{I}$ which can be purchased under this pricing.

Theorem 4 follows from the following two lemmas.

Lemma 2. *Algorithm* TB *runs in quasi-polynomial time in m, for any fixed $\epsilon > 0$.*

Lemma 3. *For any $\epsilon > 0$, Algorithm* TB *returns a pricing p and a set of intervals $\mathcal{J}$ such that $p(I) \leq B(I)$ for all $I \in \mathcal{J}$ and $p(\mathcal{J}) \geq (1 - 3\epsilon)p^*(\text{OPT})$.*

5 Hardness of the Highway Problem

5.1 Strong NP-Hardness in the Standard Model

Recall that the highway problem is the special case of the tollbooth problem when the underlying graph is a path. In this section, we show that the problem is strongly NP-hard, thus ruling out the existence of an FPTAS for the problem. Consider a MAX-2-SAT instance with n variables $\{x_1, \dots, x_n\}$ and m clauses $\{C_1, \dots, C_m\}$. Let the variables be numbered $1, \dots, n$.

Theorem 5. *The highway problem is strongly NP-hard.*

Proof. The proof follows by the construction of gadgets for the variables and clauses in a given MAX-2-SAT instance. We next describe their construction.
Variable Gadget: The gadget for each variable consists of two copies of the following *basic gadget*, and a *consistency gadget*.
Basic Gadget: The basic gadget consists of 4 edges $e_1, \dots, e_4$, and 4 types of intervals A, B, C and D. There are 4 intervals each of type A and B, labeled $a_1, \dots, a_4$, and $b_1, \dots, b_4$ respectively. The intervals $a_i = b_i = [e_i]$, for $i = 1, \dots, 4$. The intervals $a_1, \dots, a_4$ have budgets of $1, 2, 2, 1$ respectively, and the intervals $b_1, \dots, b_4$ have budgets $2, 1, 1, 2$ respectively. There are 2 type C intervals, c_1 and c_2, with $c_1 = [e_1, e_2]$, and $c_2 = [e_3, e_4]$. These intervals have a budget of 3. There are two intervals of type D, $d_1 = d_2 = [e_2, e_3]$ with d_1 having a budget of 4, and d_2, a budget of 2. The basic gadget is shown in Figure 1. We can show that there are exactly two price assignments for $\{e_1, \dots, e_4\}$ that gives us optimum profit.

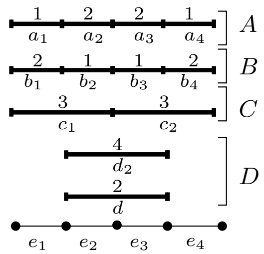


Fig. 1. A basic gadget. The gadget consists of 4 edges, and 4 types of intervals A, B, C and D. The interval labels are shown below each interval, and the budgets are shown above each interval.

Lemma 4. *The maximum profit that can be obtained from a basic gadget is* 18*, and there are exactly two sets of prices namely* $(1, 2, 2, 1)$ *and* $(2, 1, 1, 2)$ *for the edges* $(e_1, \ldots, e_4)$ *that achieve this profit.*

The price assignment $(1, 2, 2, 1)$ and $(2, 1, 1, 2)$ to the edges $e_1, \ldots, e_4$ respectively are called TRUE and FALSE assignments respectively. The variable gadget is constructed on $8n + 1$ edges $(e_{4n}, e_{4n-1}, \ldots, e_1, h, f_1, \ldots, f_{4n})$, where n is the number of variables in the MAX-2-SAT instance. Each variable gadget consists of two copies of the basic gadget, along with a consistency gadget. The consistency gadget ensures that the two basic gadgets have the same price assignment. More formally, let $(x_1, \ldots, x_n)$ be an order on the variables of the MAX-2-SAT instance. Then, the gadget for variable x_i consists of two basic gadgets, B_i^1 and B_i^2. B_i^1 consists of intervals (customers) interested in the edges $e_{4i-3}, \ldots, e_{4i}$ and B_i^2 consists of intervals interested in the edges $f_{4i-3}, \ldots, f_{4i}$. Finally, the intervals ensuring consistency of the gadget for variable x_i spans from $e_{4i-1}, \ldots, f_{4i-3}$. The consistency gadget consists of a single interval that has a budget of $mn^2 + 6(2i - 2) + 6$. Finally, we add a new type of interval, called a type H interval that is interested only in the edge h, and has a budget of mn^2.

Figure 2 shows the arrangement of the variable gadgets. We can show that the consistency intervals do their job, i.e., if for a variable gadget, B_i^1 and B_i^2 have different price assignments, we obtain a smaller profit than when they are the same.

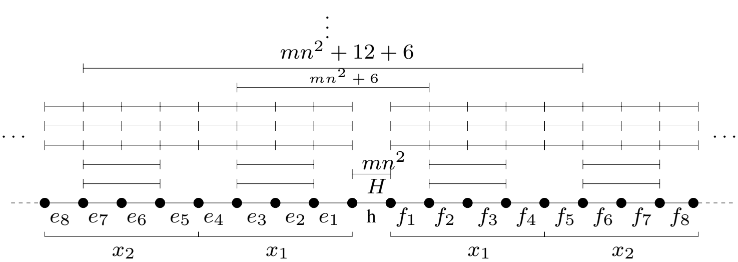


Fig. 2. The variable gadget

Lemma 5. *The maximum profit of* $2mn^2 + 6(2i - 2) + 6 + 36$ *from a variable gadget and the interval* h *is achieved only when both the basic gadgets corresponding to a variable are consistent, and the type* H *interval purchases edge* h *at a price of* mn^2.

We will create several copies of the basic gadgets, the consistency gadgets for each variable as well as several copies of the H interval to ensure that in an optimum price assignment, the basic gadgets are consistent, and the reduction goes through. But before we do this, we describe the clause gadgets.

Clause Gadgets: The clause gadget for a clause of variables x_i and x_j runs between the basic gadget B_i^1 and B_j^2. There are four types of clause gadgets corresponding to the four types of clauses. Each clause gadget consists of one interval. These intervals have the property that we obtain a certain revenue from the clause interval if and only if the clause is satisfied; otherwise we obtain nothing. (See the table in Figure 3).

Clause	Interval	Budget
$(x_i \vee x_j)$	$[e_{4i-3}, f_{4j-3}]$	$mn^2 + 6(i+j-2) + 3$
$(\overline{x_i} \vee x_j)$	$[e_{4i-1}, f_{4j-3}]$	$mn^2 + 6(i+j-2) + 6$
$(x_i \vee \overline{x_j})$	$[e_{4i-3}, f_{4j-1}]$	$mn^2 + 6(i+j-2) + 6$
$(\overline{x_i} \vee \overline{x_j})$	$[e_{4i-1}, f_{4j-1}]$	$mn^2 + 6(i+j-2) + 9$

Fig. 3. This table shows the lengths and budgets of the intervals making up a clause gadget for the four different kinds of clauses

We say that a pricing is *consistent* if for every variable, the price assignment to the two basic gadgets of the variable gadget are both TRUE or both FALSE, and the consistency intervals spend their entire budgets.

Lemma 6. *Consider a clause* C *consisting of variables* x_i *and* x_j *and a consistent price assignment to the edges. Then, the intervals corresponding to* C *will be able to purchase their desired edges if and only if the corresponding truth assignment to the variables satisfies the clause* C.

We now describe the final reduction. As mentioned earlier, we have to create copies of the variable gadget, consistency gadget and the H interval for the proof to go through. We make T copies of each basic gadget, of each consistency gadget, and of the H interval, where any value of T, larger than m^2n^3 will suffice for the proof. Observe that for a variable gadget again, the profit maximizing prices achieve consistency of the variable gadget, and making T copies of the H intervals ensures that the price of the edge h is set to mn^2.

5.2 APX-Hardness in the Discount Model

When negative prices are allowed, the highway problem becomes APX-hard.

Theorem 6. *The highway problem with negative prices is APX-hard even when restricted to instances in which one edge is shared by all the customers.*

We prove the above theorem by first showing that it is equivalent to a pricing problem on bipartite graphs and then prove that the latter is APX-hard via a reduction from maxcut on 3-regular graphs. The details of the proof can be found in the extended version of this paper. This result has been independently obtained in [18].

6 Conclusion

We presented an $O(\log n)$-approximation algorithm for the tollbooth problem on trees, which is better than the current upper bound for the general problem. Improving this bound is an interesting open problem. One plausible direction towards this is to use as a subroutine, the quasi-polynomial time algorithm for the case of uncrossing paths. Such techniques have been used before, for example for the multicut problem on trees [14]. However, it is unclear how a general instance of the Tb problem can be decomposed into a set of problems of the uncrossing type. For the highway problem, the strong NP-hardness presented in this paper shows that the problem is almost closed, modulo improving the running time from quasi-polynomial to polynomial. When negative prices are allowed, somewhat surprisingly, the problem becomes harder, and even a quasi-PTAS is unlikely to exist.

Acknowledgements. We would like to thank Naveen Garg for suggesting to use the separator theorem in the proof of Theorem 1.

References

1. Aggarwal, G., Hartline, J.D.: Knapsack auctions. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 1083–1092. ACM Press, New York (2006)
2. Balcan, M.F., Blum, A.: Approximation algorithms and online mechanisms for item pricing. In: EC 2006: Proceedings of the 7th ACM conference on Electronic commerce, pp. 29–35. ACM Press, New York (2006)
3. Balcan, M.-F., Blum, A., Chan, H., Hajiaghayi, M.: A theory of loss-leaders: Making money by pricing below cost. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 293–299. Springer, Heidelberg (2007)
4. Balcan, M.F., Blum, A., Mansour, Y.: Item pricing for revenue maximization. In: EC 2008: Proceedings of the 9th ACM conference on Electronic commerce. ACM Press, New York (to appear) (2008)
5. Balcan, M.F., Blum, A.: Approximation algorithms and online mechanisms for item pricing. Theory of Computing 3, 179–195 (2007)
6. Briest, P.: Uniform budgets and the envy-free pricing problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 808–819. Springer, Heidelberg (2008)

7. Briest, P., Hoefer, M., Krysta, P.: Stackelberg network pricing games. In: Proc. 25th International Symposium on Theoretical Aspects of Computer Science (STACS), pp. 133–142 (2008)
8. Briest, P., Krysta, P.: Single-minded unlimited supply pricing on sparse instances. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 1093–1102. ACM Press, New York (2006)
9. Briest, P., Krysta, P.: Buying cheap is expensive: Hardness of non-parametric multi-product pricing. In: Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM (2007)
10. Cheung, M., Swamy, C.: Approximation algorithms for single-minded envy-free profit-maximization problems with limited supply. In: FOCS 2008, pp. 35–44 (2008)
11. Demaine, E.D., Hajiaghayi, M.T., Feige, U., Salavatipour, M.R.: Combination can be hard: approximability of the unique coverage problem. In: SODA 2006: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, pp. 162–171. ACM Press, New York (2006)
12. Elbassioni, K.M., Sitters, R.A., Zhang, Y.: A quasi-PTAS for profit-maximizing pricing on line graphs. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 451–462. Springer, Heidelberg (2007)
13. Glynn, P.W., Van Roy, B., Rusmevichientong, P.: A nonparametric approach to multi-product pricing. Operations Research 54(1), 82–98 (2006)
14. Golovin, D., Nagarajan, V., Singh, M.: Approximating the k-multicut problem. In: SODA 2006, pp. 621–630 (2006)
15. Grigoriev, A., van Loon, J., Sitters, R., Uetz, M.: How to sell a graph: Guidelines for graph retailers. In: Fomin, F.V. (ed.) WG 2006. LNCS, vol. 4271, pp. 125–136. Springer, Heidelberg (2006)
16. Guruswami, V., Hartline, J.D., Karlin, A.R., Kempe, D., Kenyon, C., McSherry, F.: On profit-maximizing envy-free pricing. In: SODA 2005: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, pp. 1164–1173. Society for Industrial and Applied Mathematics (2005)
17. Hartline, J.D., Koltun, V.: Near-optimal pricing in near-linear time. In: Dehne, F., López-Ortiz, A., Sack, J.-R. (eds.) WADS 2005. LNCS, vol. 3608, pp. 422–431. Springer, Heidelberg (2005)
18. Khandekar, R., Kimbrel, T., Makarychev, K., Sviridenko, M.: On hardness of pricing items for single-minded bidders. In: Proceedings, 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX (2009)
19. Luby, M., Wigderson, A.: Pairwise independence and derandomization. Foundations and Trends in Theoretical Computer Science 1(4), 237–301 (2005)
20. Motwani, R., Raghavan, P.: Randomized algorithms. Cambridge University Press, Cambridge (1995)

On the Complexity of Iterated Weak Dominance in Constant-Sum Games

Felix Brandt, Markus Brill, Felix Fischer, and Paul Harrenstein

Institut für Informatik, Ludwig-Maximilians-Universität München
80538 München, Germany
{brandtf,brill,fischerf,harrenst}@tcs.ifi.lmu.de

Abstract. In game theory, a player's action is said to be weakly dominated if there exists another action that, with respect to what the other players do, is never worse and sometimes strictly better. We investigate the computational complexity of the process of iteratively eliminating weakly dominated actions (IWD) in two-player constant-sum games, i.e., games in which the interests of both players are diametrically opposed. It turns out that deciding whether an action is eliminable via IWD is feasible in polynomial time whereas deciding whether a given subgame is reachable via IWD is NP-complete. The latter result is quite surprising as we are not aware of other natural computational problems that are intractable in constant-sum games. Furthermore, we slightly improve a result by Conitzer and Sandholm [6] by showing that typical problems associated with IWD in win-lose games with at most one winner are NP-complete.

1 Introduction

A simple and indisputable conviction in game theory is that a player need not bother to consider an action that yields less payoff than some other action *no matter* what all the other players do (see, e.g., [12]). In game-theoretic terms, such an action is *strictly dominated*. Similarly, one says that an action is *weakly dominated* if there exists another action that, with respect to what the other players do, is never worse and sometimes strictly better. An action that is not weakly dominated is also said to be *admissible*. When a (strictly or weakly) dominated action is eliminated from a player's consideration, it may be possible that a previously undominated action of another player becomes dominated. Thus, based on the mutual rational belief that (some) dominated actions will not be played, one can define an iterative process of eliminating actions. It is well-known that this process invariably leads to the same subgame no matter in which order strictly dominated actions are eliminated whereas this is not the case for weak dominance (see, e.g., [1, 20]). The dependence on the order of elimination gives rise to some combinatorial difficulties as witnessed by the NP-completeness of various computational problems related to iterated weak dominance [6, 8]. By contrast, the corresponding problems for iterated strict dominance are computationally tractable. This disparity has also become apparent in the complexity analysis of other solution concepts based on dominance [4].

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 287–298, 2009.

We investigate the computational complexity of iterated weak dominance (IWD)—or *iterated admissibility*—in *two-player constant-sum* games, i.e., games in which the interests of both players are diametrically opposed. Our analysis is restricted to dominance by pure strategies although most of our results can be readily applied to mixed strategies as well (see Section 6). It turns out that deciding whether an action is eliminable via IWD is feasible in polynomial time whereas deciding whether a given subgame is reachable via IWD is NP-complete. The latter result is quite surprising as we are not aware of other natural computational problems that are intractable in normal-form constant-sum games.[1] Furthermore, we slightly improve a result by Conitzer and Sandholm [6] by showing that typical problems associated with IWD in win-lose games *with at most one winner* are NP-complete.

Iterated weak and strict dominance are well-established solution concepts, which have a long history and occur in virtually every textbook on game theory. The publication of Bernheim [2] and Pearce [17] instigated a renewed discussion concerning their formal and intuitive connections with rationalizability and the epistemic foundations of solution concepts [3, 18], the stability of equilibria [10], and backward induction solutions [7, 19]. It cannot be said that iterated weak dominance has left the arena entirely unscathed. Unlike iterated strict dominance, proper epistemic foundations for iterated weak dominance are pretty hard to come by. In particular, Samuelson [18] showed that common knowledge of admissibility does not imply iterated weak dominance. Nevertheless, IWD has been argued to have its place as a tool in the analysis of games (see, e.g., [14, 15], for discussions). Our aim, however, is by no means to pass judgement on iterated weak dominance as a solution concept as such. Rather, our focus is on the computational aspects of IWD in two-player zero-sum and win-lose games with at most one winner. As mentioned above, the fact that some of these problems turn out to be NP-hard is interesting and surprising in its own right.

After having introduced our formal framework (Section 2), we introduce the auxiliary concept of a regionalized game in Section 3. We prove that regionalized games may be used as a convenient tool in the proofs of our hardness results. In Section 4 we deal with the computational complexity of the reachability and eliminability problems in two-player constant-sum games. Finally, we address the same problems for win-lose games that allow at most one winner in Section 5. Due to space restrictions, some of the proofs are omitted.

2 Preliminaries

A *two-player game* $\Gamma = (A_1, A_2, u)$ is given by a finite set A_1 of actions of player 1, a finite set A_2 of actions of player 2, and a utility function $u : A_1 \times A_2 \to \mathbb{R} \times \mathbb{R}$. We also have A denote $A_1 \cup A_2$ and write $u_1(a, b) = x$ and $u_2(a, b) = y$ if $u(a, b) = (x, y)$. Both players are assumed to choose one of their

[1] Hard problems are known in the context of *extensive* constant-sum games. For instance, Koller and Megiddo [11] show that finding maximin behavior strategy in constant-sum extensive form games without perfect recall is NP-hard.

actions simultaneously. If player 1 chooses a and player 2 chooses b, their payoffs will be $u_1(a,b)$ and $u_2(a,b)$, respectively.

A two-player game is called *constant-sum* if $u_1(a,b) + u_2(a,b) = u_1(c,d) + u_2(c,d)$ for all $a,c \in A_1$ and $b,d \in A_2$. It is convenient to write down the payoffs of a game in a matrix with rows indexed by the actions of player 1 and columns indexed by the actions of players 2.

Consider a game and let $a,b \in A_1$ be two actions of player 1. Then a is said to *weakly dominate* b *at* $c \in A_2$ if $u_1(a,c) > u_1(b,c)$ and for all $d \in A_2$, $u_1(a,d) \geq u_1(b,d)$. More generally, a is said to weakly dominate b if a weakly dominates b at c for some $c \in A_2$. We further say that $c \in A_2$ *backs* the elimination of b by a if $u_1(a,c) > u_1(b,c)$, and *blocks* the elimination of b by a if $u_1(a,c) < u_1(b,c)$. Dominance, backing, and blocking for actions of player 2 is defined analogously. Obviously an action is dominated by another action of the same player if some action of the other player backs the elimination, and none of them blocks it. As the remainder of this paper only concerns (iterated) weak dominance, we will drop the qualification 'weak' and by 'dominance' understand weak dominance, unless stated otherwise.

An *elimination sequence* of a game $\Gamma = (A_1, A_2, u)$ is a finite sequence $\Sigma = (\Sigma_1, \ldots, \Sigma_k)$ of subsets of actions in $A = A_1 \cup A_2$. For a game $\Gamma = (A_1, A_2, u)$ and an elimination sequence $\Sigma = (\Sigma_1, \ldots, \Sigma_k)$ of Γ we have $\Gamma(\Sigma)$ denote the subgame where the actions in $\Sigma_1 \cup \cdots \cup \Sigma_k$ have been removed, i.e., $\Gamma(\Sigma) = (A'_1, A'_2, u')$ where $A'_1 = A_1 \setminus (\Sigma_1 \cup \cdots \cup \Sigma_k)$ and $A'_2 = A_2 \setminus (\Sigma_1 \cup \cdots \cup \Sigma_k)$ and u' is the restriction of u to $A'_1 \times A'_2$. The *validity* of elimination sequences is then defined inductively: the empty sequence ϵ is valid for every game and an elimination sequence $(\Sigma_1, \ldots, \Sigma_m, \Sigma_{m+1})$ is valid in Γ if $(\Sigma_1, \ldots, \Sigma_m)$ is valid in Γ and every action $a \in \Sigma_{m+1}$ is dominated in $\Gamma(\Sigma_1, \ldots, \Sigma_m)$. If in $(\Sigma_1, \ldots, \Sigma_m)$ for each $1 \leq i \leq m$, Σ_i is a singleton, we say the elimination sequence is *simple*. Simple elimination sequences we usually write as sequences $\sigma = (\sigma_1, \ldots, \sigma_m)$ of actions in A.

Let Γ be a game. Then, an action a is called *eliminable by* b *at* c in Γ if there exists a valid elimination sequence Σ such that a is dominated by b at c in $\Gamma(\Sigma)$. Action a is *eliminable* in Γ if there are actions b and c such that a is eliminable by b at c. A subgame Γ' of Γ is *reachable* from Γ if there exists a valid elimination sequence Σ such that $\Gamma(\Sigma) = \Gamma'$. Furthermore Γ is called *solvable* if some game $\Gamma' = (A'_1, A'_2, u')$ with $|A'_1| = |A'_2| = 1$ is reachable from Γ. Finally, we say Γ is *irreducible* if none of its actions is dominated.

We assume familiarity with the theory of complexity, in particular with the complexity classes P and NP and the canonical problem *3SAT* (see, e.g., [16]).

3 Regions and Regionalized Games

An essential building block of our hardness proofs are *regionalized games*.

Definition 1. *A* regionalized two-player game *is a tuple* (Γ, X_1, X_2) *consisting of a two-player game* $\Gamma = (A_1, A_2, u)$ *and partitions* X_1 *and* X_2 *of* A_1 *and* A_2*, respectively. The elements of* X_1 *and* X_2 *are also called* regions.

For regionalized games the concept of a valid elimination sequence is modified, so as to allow only eliminations of actions that are dominated by other actions in the same region.

Definition 2. *A* valid elimination sequence *for a regionalized game* (Γ, X_1, X_2) *is a sequence* $\Sigma = (\Sigma_1, \ldots, \Sigma_k)$ *for* Γ *such that for each* i *with* $1 \leq i \leq k$ *and each* $a \in \Sigma_i$*, there is some action* b *and some* $x \in X_1 \cup X_2$ *such that* $a, b \in x$ *and* b *dominates* a *in* $\Gamma(\Sigma_1, \ldots, \Sigma_{i-1})$*.*

The following lemma shows that any regionalized two-player game can be transformed in polynomial time into a non-regionalized two-player game with the same valid elimination sequences. The significance of this result is that for the computational problems we consider—reachability of (irreducible) subgames, eliminability and solvability—we can restrict ourselves to regionalized games, which are often more practical for and afford more insight into the constructions used in our hardness proofs than games without regions.

Lemma 1. *For each regionalized game* (Γ, X_1, X_2) *with* $\Gamma = (A_1, A_2, u)$*, there is a game* $\Gamma' = (A_1', A_2', u')$ *computable in polynomial time such that the valid elimination sequences of* Γ' *and* (Γ, X_1, X_2) *coincide:*

$$\{\Sigma \colon \Sigma \text{ a valid sequence in } \Gamma'\} = \{\Sigma \colon \Sigma \text{ a valid sequence in } (\Gamma, X_1, X_2)\}.$$

Moreover, $u'(a, b) \in \{(0, 1), (1, 0)\}$ *for all* $a \in A_1' \setminus A_1$ *and* $b \in A_2' \setminus A_2$*.*

4 Two-Player Constant-Sum Games

We show that subgame reachability is NP-complete even in games that only allow the outcomes $(0, 1)$ and $(1, 0)$. This may be attributed to the order dependence of IWD. In Section 4.2 we find that for two-player constant-sum games a weak form of order independence can be salvaged, which allows us to formulate an efficient algorithm for the eliminability problem. We first show that in the case of two-player zero-sum games we can restrict our attention to *simple* elimination sequences.

Lemma 2. *Let* Γ *be a two-player constant-sum game and* $\Sigma = (\Sigma_1, \ldots, \Sigma_m)$ *a valid elimination sequence. Then, there is a simple elimination sequence* $\sigma = (\sigma_1, \ldots, \sigma_k)$ *with* $\{\sigma_1, \ldots, \sigma_m\} = \Sigma_1 \cup \cdots \cup \Sigma_m$ *that is also valid in* Γ*.*

Since every simple elimination sequence is an elimination sequence, it follows that a subgame of a two-player zero-sum game is reachable if and only if it is reachable by a simple elimination sequence. Analogous statements hold for eliminability and solvability in two-player zero-sum games.

Lemma 2 does not hold for general strategic games. In particular it fails for games with outcomes in $\{(0, 0), (0, 1)(1, 0)\}$, as the game in Figure 1 shows.

	y	y'
x	$(0,0)$	$(0,1)$
x'	$(1,0)$	$(0,0)$

Fig. 1. Both x and y are weakly dominated in the game above. Hence, the elimination sequence $\{x, y\}$ is valid. However, neither of the simple elimination sequences (x, y) and (y, x) is valid.

4.1 Reachability

We first show that subgame reachability in constant-sum games is intractable.

Theorem 1. *Given constant-sum games Γ and Γ', deciding whether Γ' is reachable from Γ is NP-complete, even when restricted to outcomes* $(0,1)$ *and* $(1,0)$ *and Γ' is to be irreducible.*

Proof. For membership in NP consider arbitrary constant-sum games Γ and Γ'. Given an elimination sequence $\sigma = (\sigma_1, \dots, \sigma_k)$, it can clearly be decided in polynomial time whether σ is a valid elimination sequence for (Γ, X_1, X_2) such that $\Gamma' = \Gamma(\sigma)$.

The proof of hardness proceeds by a reduction from *3SAT*. By virtue of Lemma 1 it suffices to prove this for regionalized games. Consider an arbitrary *3CNF*$\varphi = C_1 \wedge \cdots \wedge C_k$, where each $C_i = (\lambda_i^1 \vee \lambda_i^2 \vee \lambda_i^3)$ is a clause and each λ_i^j is a literal, for $1 \leq i \leq k$ and $1 \leq j \leq 3$. Define the regionalized game $(\Gamma_\varphi, X_1, X_2)$, with $\Gamma_\varphi = (A_1, A_2, u)$ as follows.

$$
\begin{aligned}
A_1 &= \{p, \neg p, \neg(p \wedge \neg p) \colon p \text{ a variable in } \varphi\} \\
&\quad \cup \{C_i, (\lambda_i^1, i), (\lambda_i^2, i), (\lambda_i^3, i) \colon C_i \text{ a clause in } \varphi\} \\
&\quad \cup \{e\} \\
A_2 &= \{p, \neg p \colon p \text{ a variable in } \varphi\} \cup \{a, b\} \\
X_1 &= \{\{p, \neg p, \neg(p \wedge \neg p)\} \colon p \text{ a variable in } \varphi\} \\
&\quad \cup \{\{C_i, (\lambda_i^1, i), (\lambda_i^2, i), (\lambda_i^3, i)\} \colon C_i \text{ a clause in } \varphi\} \\
&\quad \cup \{\{e\}\} \\
X_2 &= \{\{p, \neg p \colon p \text{ a variable in } \varphi\} \cup \{a, b\}\} = \{A_2\}
\end{aligned}
$$

For each propositional variable p occurring in φ, the payoffs in rows p, $\neg p$ and $\neg(p \wedge \neg p)$ are defined as in the following table, where q is a typical variable in φ distinct from p.

	p	$\neg p$	...	q	$\neg q$	...	a	b
p	$(1,0)$	$(0,1)$		$(0,1)$	$(0,1)$		$(1,0)$	$(0,1)$
$\neg p$	$(0,1)$	$(1,0)$		$(0,1)$	$(0,1)$		$(1,0)$	$(0,1)$
$\neg(p \wedge \neg p)$	$(1,0)$	$(1,0)$		$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$

Due to the regionalization, $\neg(p \wedge \neg p)$ can be eliminated only by row p or row $\neg p$. Moreover, column a is the only action backing the elimination of $\neg(p \wedge \neg p)$. Also, at least one of the columns p and $\neg p$ needs to be removed (by column b) before $\neg(p \wedge \neg p)$ can be eliminated. Intuitively, removing column p means setting variable p to false, removing column $\neg p$, setting variable p to true, thus choosing a valuation.

Also for each i with $1 \leq i \leq k$, the payoffs in rows C_i, (λ_i^1, i), (λ_i^2, i), (λ_i^3, i) depend on the literals occurring in C_i. In the table below, $\overline{\lambda}_i^j = \neg p$, if $\lambda_i^j = p$, and $\overline{\lambda}_i^j = p$, if $\lambda_i^j = \neg p$. Also, we assume $i \neq m$.

	λ_i^1	$\overline{\lambda}_i^1$	λ_i^2	$\overline{\lambda}_i^2$	λ_i^3	$\overline{\lambda}_i^3$		λ_m^j	$\overline{\lambda}_m^j$		a	b
(λ_i^1, i)	$(1,0)$	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$
(λ_i^2, i)	$(0,1)$	$(0,1)$	$(1,0)$	$(0,1)$	$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$
(λ_i^3, i)	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(1,0)$	$(0,1)$		$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$
C_i	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$		$(0,1)$	$(0,1)$		$(1,0)$	$(0,1)$

Thus, the only columns backing the elimination of C_i are λ_i^1, λ_i^2 and λ_i^3. Also notice that column a blocks the elimination of C_i. Nevertheless, as we saw above, column a is essential to the elimination of the rows $\neg(p \wedge \neg p)$. Intuitively, this signifies that a valuation needs to be chosen before any of the rows C_i is eliminated.

Finally, let $u(e, y) = (1, 0)$ if $y \neq b$, and $u(e, y) = (0, 1)$.

	λ_1^1	$\overline{\lambda}_1^1$	$\cdots$	λ_k^3	$\overline{\lambda}_k^3$	a	b
e	$(1,0)$	$(1,0)$	$\cdots$	$(1,0)$	$(1,0)$	$(1,0)$	$(0,1)$

Observe that row e is the only action in its region and as such cannot be eliminated. Also, e backs the elimination of every column by b.

Now define $(\Gamma'_\varphi, X'_1, X'_2)$ with $\Gamma'_\varphi = (A'_1, A'_2, u')$ such that

$$\begin{aligned} A'_1 &= \{p, \neg p \colon p \text{ a variable in } \varphi\} \\ A'_2 &= \{b\}. \end{aligned}$$

Moreover, we have u', X'_1 and X'_2 appropriately restricted to A'_1 and A'_2, i.e., $u' = u|_{A'_1 \times A'_2}$, $X'_1 = \{x \cap A'_1 \colon x \in X_1\}$ and $X'_2 = \{x \cap A'_2 \colon x \in X_2\}$. It is easily appreciated that in $(\Gamma'_\varphi, X'_1, X'_2)$ there are no actions that can be eliminated, i.e., $(\Gamma'_\varphi, X'_1, X'_2)$ is irreducible.

We now prove that φ is satisfiable if and only if $(\Gamma'_\varphi, X'_1, X'_2)$ is reachable from $(\Gamma_\varphi, X_1, X_2)$.

Assume that φ is satisfiable and let v be a valuation satisfying φ. Now let b eliminate all columns representing a literal λ that is set to false by v.

Subsequently, being backed by column a, for each variable p, row p or row $\neg p$, as the case may be, eliminates row $\neg(p \wedge \neg p)$. Next, a itself is eliminated by b, removing the blocks at a on C_i for each i with $1 \leq i \leq k$. Having assumed φ to be satisfiable, for each clause C_i there still is a column λ_i^j present. Backed by this column, row C_i can now be eliminated by row (λ_i^j, i). All rows $\neg(p \wedge \neg p)$ for a variable p and C_i for $1 \leq i \leq k$ being removed, column b eliminates all remaining columns, thus reaching subgame $(\Gamma'_\varphi, X'_1, X'_2)$.

For the opposite direction assume that $(\Gamma'_\varphi, X'_1, X'_2)$ is reachable from (Γ, X_1, X_2) and let σ be the witnessing elimination sequence. Now observe that for each variable p occurring in φ row $\neg(p \wedge \neg p)$ is eliminated. Recall that this is only possible when at least one of the columns p and $\neg p$ is eliminated first and when column a is still present to back the elimination. Also, for each $1 \leq i \leq k$ row C_i is eliminated in σ. This elimination, however, is only possible by some row (λ_i^j, i) backed by column λ_i^j, and only when column a is no longer there to block it. Now define a valuation v^* such that v^* satisfies all literals λ_i^j represented by columns that are still present at the point that column a is eliminated in σ. It follows that v^* is well-defined and also satisfies φ. □

By definition, solvability is a special case of subgame reachability, which Theorem 1 shows to be intractable in constant-sum games. For single-winner games, i.e., constant-sum games consisting only of the outcomes $(0, 1)$ and $(1, 0)$, this problem is tractable [5]. Whether solvability is tractable in general constant-sum games, however, remains an open question.

4.2 Eliminability

The failure of IWD being order independent, can be rephrased as that, for elimination sequences $\sigma = (\sigma_1, \dots, \sigma_k)$ and actions d, σ being valid in a game Γ does not generally imply that σ is still valid in $\Gamma(d)$ (or that $(\sigma_1, \dots, \sigma_{k-1}, \sigma_{k+1}, \dots, \sigma_n)$ is, if $d = \sigma_k$). The problem is that there may be i with $1 \leq i \leq n$ such that action d is the only action in $\Gamma(\sigma_1, \dots, \sigma_{i-1})$ that backs the elimination of σ_i. Eliminating d too early may thus render σ_i uneliminable. For two-player constant-sum games, we find, however, that under particular circumstances and for a particular type of elimination sequence, which we will call *essential*, one can carry out the elimination of d earlier and still be able to eliminate all of the actions $\sigma_1, \dots, \sigma_n$, provided one is prepared to postpone the elimination of some of them. This observation forms the basis of Theorem 2, proving that the eliminability problem for two-player constant-sum games can be solved efficiently.

Fix a game Γ and let $\sigma = (\sigma_1, \dots, \sigma_n)$ and $\delta = (\delta_1, \dots, \delta_n)$ be sequences of actions. We say that σ *is valid in* Γ *with respect to* δ if for each i with $1 \leq i \leq n$, action δ_i dominates σ_i in $\Gamma(\sigma_1, \dots, \sigma_{i-1})$. This implies that $\delta_i \notin \{\sigma_1, \dots, \sigma_i\}$. Given σ and δ we also define for each i with $1 \leq i \leq k$,

$$B_i(\delta, \sigma) = \{(\delta_j, \sigma_j) \colon \sigma_i \text{ blocks the elimination of } \sigma_j \text{ by } \delta_j \text{ in } \Gamma\}.$$

Observe that $j > i$ for all $(\delta_j, \sigma_j) \in B_i(\delta, \sigma)$, if σ is valid in Γ with respect to δ. Also, if $B_i(\delta, \sigma) = \emptyset$ for some i with $1 \leq i \leq n$ the elimination sequence $(\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$ is also valid in Γ. We say that σ is *essential* if $B_i(\delta, \sigma) \neq \emptyset$ for all i with $1 \leq i < n$. An action σ_i is said to an *obstacle* in σ with respect to δ if δ_i does dominate σ_i in $\Gamma(\sigma_1, \ldots, \sigma_{i-1})$. The set of obstacles of σ with respect to δ we denote by $O(\delta, \sigma)$. Finally, for $\sigma = (\sigma_1, \ldots, \sigma_n)$ a sequence of actions and $1 \leq i < j \leq n$, we write

$$\sigma^{i \to j} = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_j, \sigma_i, \sigma_{j+1}, \ldots, \sigma_n).$$

Thus, $\sigma^{i \to j}$ is exactly like σ with the only difference that σ_i moved to the position directly behind σ_j. We now have the following useful lemma, which specifies sufficient conditions under which the elimination of an action can be delayed without producing new obstacles apart from the action itself.

Lemma 3. *Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ and $\delta = (\delta_1, \ldots, \delta_n)$ be sequences of actions of a game $\Gamma = (A_1, A_2, u)$. Fix an i with $1 \leq i \leq n$ and let m be the smallest index k with $i < k \leq n$ such that $(\delta_k, \sigma_k) \in B_i(\delta, \sigma)$. Then, for all j with $i \leq j < m$, $O(\delta^{i \to j}, \sigma^{i \to j}) \subseteq O(\sigma, \delta) \cup \{\sigma_i\}$.*

Proof. Consider an arbitrary σ_k with $1 \leq k \leq n$ and $k \neq i$. Assume that σ_k is no obstacle in σ with respect to δ, i.e., σ_k be dominated by δ_k at some action x in $\Gamma(\sigma_1, \ldots, \sigma_{k-1})$. The only interesting case is $i < k \leq j$, as otherwise $\Gamma(\sigma_1, \ldots, \sigma_{k-1}) = \Gamma(\sigma_1^{i \to j}, \ldots, \sigma_{k-1}^{i \to j})$. Hence, $\sigma_k = \sigma_{k-1}^{i \to j}$ and $(\delta_k, \sigma_k) \notin B_i(\delta, \sigma)$. Observe that in $\Gamma(\sigma_1^{i \to j}, \ldots, \sigma_{k-1}^{i \to j})$, action x still backs the elimination of σ_k by δ_k. As $(\delta_k, \sigma_k) \notin B_i(\delta, \sigma)$, σ_i does not block this elimination, neither do any other actions in $\Gamma(\sigma_1^{i \to j}, \ldots, \sigma_{k-1}^{i \to j})$. Therefore, σ_k is no obstacle in $\sigma^{i \to j}$. □

One corollary of Lemma 3 is that, under the conditions specified, $\sigma^{i \to j}$ is valid in Γ if σ is. Furthermore, if an obstacle σ_i in σ is moved to a position j where it is no longer an obstacle, and j is smaller than the smallest index with $(\delta_k, \sigma_k) \in B_i(\delta, \sigma)$ but greater than i, the number of obstacles strictly decreases, i.e., $|O(\delta^{i \to j}, \sigma^{i \to j})| < |O(\delta, \sigma)|$. We now have the following lemma.

Lemma 4. *Let $\Gamma = (A_1, A_2, u)$ be a constant-sum game. Let a, b and c be distinct actions in $A_1 \cup A_2$ and σ a valid elimination sequence in Γ containing neither a, b nor c. Then, if a is eliminable by b at c in Γ, a is still eliminable by b at c in $\Gamma(\sigma)$.*

Proof. Let d be an action distinct from a, b and c that is dominated by some action x in Γ. It suffices to prove the following:

If a is eliminable by b at c in Γ, then a is eliminable by b at c in $\Gamma(d)$.

The lemma then follows by a straightforward induction.

Assume a is eliminable by b at c in Γ. Accordingly, there are sequences $\sigma = (\sigma_1, \ldots, \sigma_n)$ and $\delta = (\delta_1, \ldots, \delta_n)$ in Γ such that σ is valid with respect to δ, $c \notin \{\sigma_1, \ldots, \sigma_n\}$, $\sigma_n = a$, $\delta_n = b$ and a being dominated by b at c in $\Gamma(\sigma_1, \ldots, \sigma_{n-1})$. Without loss of generality we may assume that σ is essential. We also make the following observations. First, we may assume without loss of generality that $d \neq \delta_i$ for all $1 \leq i \leq n$. Let d is dominated by action x in Γ and set for each i with $1 \leq i \leq n$

$$x_i = \begin{cases} x & \text{if } i = 1, \\ \delta_{i-1} & \text{if } x_{i-1} = \sigma_{i-1}, \\ x_{i-1} & \text{otherwise.} \end{cases}$$

Then, by transitivity of dominance, for each i with $1 \leq i \leq n$, if d dominates σ_i so does x_i and thus can go proxy for d. Second, σ_n is no obstacle in $(d, \sigma_1, \dots, \sigma_n)$. To appreciate this, observe that in $\Gamma(\sigma_1, \dots, \sigma_{n-1})$ there are no actions blocking the elimination of $\sigma_n = a$ by $\delta_n = b$, so neither are there any such actions in $\Gamma(d, \sigma_1, \dots, \sigma_{n-1})$. Moreover, c still backs the elimination of $\sigma_n = a$ by $\delta_n = b$. Hence, $\sigma_n = a$ is dominated by $\delta_n = b$ at c in $\Gamma(d, \sigma_1, \dots, \sigma_{n-1})$.

We consider the case in which $d \notin \{\sigma_1, \dots, \sigma_n\}$; apart from some tedious details, the case in which $d \in \{\sigma_1, \dots, \sigma_n\}$ runs along analogous lines. We show by induction on the number of obstacles in $(d, \sigma_1, \dots, \sigma_n)$ with respect to $(x, \delta_1, \dots, \delta_n)$ that there is a sequence $\tau = (\tau_1, \dots, \tau_n)$ that is valid in $\Gamma(d)$ and, moreover, is such that b dominates a at c in $\Gamma(d, \tau_1, \dots, \tau_{n-1})$.

If $(d, \sigma_1, \dots, \sigma_n)$ contains no obstacles, σ is obviously valid in $\Gamma(d)$ and can be taken as a witness for τ. So assume $(d, \sigma_1, \dots, \sigma_n)$ contains more than one obstacle and consider the obstacle σ_i with the smallest index i. Because σ_n is no obstacle, $\sigma_i \neq \sigma_n$. Consequently, σ being both valid and essential, there is some smallest index $j > i$ such that $(\delta_j, \sigma_j) \in B_i(\delta, \sigma)$. Without loss of generality assume that $\sigma_i \in A_1$ and $\delta_j, \sigma_j \in A_2$. Then, $u_2(\sigma_i, \delta_j) < u_2(\sigma_i, \sigma_j)$ and, by Γ being constant-sum, $u_1(\sigma_i, \delta_j) > u_1(\sigma_i, \sigma_j)$. Because δ_i dominates σ_i in $\Gamma(\sigma_1, \dots, \sigma_{i-1})$ but not in $\Gamma(d, \sigma_1, \dots, \sigma_{i-1})$ and $\delta_i \neq d$, it follows that the only dominance of δ_i over σ_i in $\Gamma(\sigma_1, \dots, \sigma_{i-1})$ is at d. Consequently, $u_1(\sigma_i, x) = u_1(\delta_i, x)$ for all $x \in A_2 \setminus \{d, \sigma_1, \dots, \sigma_{n-1}\}$. In particular, $u_1(\sigma_i, \delta_j) = u_1(\delta_i, \delta_j)$ and $u_1(\sigma_i, \sigma_j) = u_1(\delta_i, \sigma_j)$. With Γ being constant-sum, it follows that $u_2(\delta_i, \delta_j) < u_2(\delta_i, \sigma_j)$, i.e., δ_i blocks the elimination of σ_j by δ_j in Γ. Accordingly, δ_i is eliminated before σ_j in σ, i.e., $\delta_i = \sigma_{i_1}$ for some $i < i_1 < j$ and $(\delta_j, \sigma_j) \in B_{i_1}(\delta, \sigma)$. Repeating this argument, there are $i = i_0 < i_1 < \cdots < i_m < j$ such that $\sigma_{i_k} = \delta_{i_{k-1}}$, $(\delta_j, \sigma_j) \in B_{i_k}(\delta, \sigma)$ for $1 \leq k \leq m$, and $u_2(\delta_{i_m}, \delta_j) \geq u_2(\delta_{i_m}, \sigma_j)$. The latter because otherwise δ_j would not dominate σ_j in $\Gamma(\sigma_1, \dots, \sigma_{j-1})$. As Γ is constant-sum, moreover, $u_1(\delta_{i_m}, \sigma_j) \geq u_1(\delta_{i_m}, \delta_j)$. By transitivity of dominance, then, $u_1(\delta_{i_m}, \delta_j) \geq u_1(\sigma_i, \delta_j)$. The situation is depicted in Figure 2. It follows that $u_1(\delta_{i_m}, \sigma_j) > u(\sigma_i, \sigma_j)$. As a consequence, δ_{i_m} dominates σ_i at σ_j in $\Gamma(d, \sigma_1, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_{i_m})$.

Now consider the elimination sequence $\sigma^{i \to i_m} = (\sigma_1^{i \to i_m}, \dots, \sigma_n^{i \to i_m})$. Recall that $i_m < j$. By choice of j and Lemma 3, the sequence $\sigma^{i \to i_m}$ is valid in Γ and $(d, \sigma_1^{i \to i_m}, \dots, \sigma_n^{i \to i_m})$ contains fewer obstacles with respect to $(x, \delta_1^{i \to i_m}, \dots, \delta_n^{i \to i_m})$ than $(d, \sigma_1, \dots, \sigma_n)$ does with respect to $(x, \delta_1, \dots, \delta_n)$. By virtue of the induction hypothesis, we may conclude that there is an elimination sequence $\tau = (\tau_1, \dots, \tau_n)$ that is valid in $\Gamma(d)$ and is such that b dominates a at c in $\Gamma(d, \tau_1, \dots, \tau_{n-1})$ □

	d	δ_j		σ_j
δ_{i_m}		$u_1(\delta_{i_m}, \delta_j)$	$\leq$	$u_1(\delta_{i_m}, \sigma_j)$
		$\geq$		$\geq$
$\delta_{i_{m-1}} = \sigma_{i_m}$		$u_1(\delta_{i_{m-1}}, \delta_j)$	$>$	$u_1(\delta_{i_{m-1}}, \delta_j)$
		$\geq$		$\geq$
$\vdots$		$\vdots$	$\vdots$	$\vdots$
		$\geq$		$\geq$
$\delta_i = \sigma_{i_1}$	$u_1(\delta_i, d)$	$u_1(\delta_i, \delta_j)$	$>$	$u_1(\delta_i, \sigma_j)$
	$\vee$	$\|$		$\|$
σ_i	$u_1(\sigma_i, d)$	$u_1(\sigma_i, \delta_j)$	$>$	$u_1(\sigma_i, \sigma_j)$

Fig. 2. Diagram illustrating the proof of Lemma 4

Intuitively, Lemma 4 says that, if one wishes to eliminate a particular action a by b backed by c, one can proceed greedily and eliminate any action whenever possible. One just has to be careful not to eliminate the actions b and c before a is eliminated. On the basis of this observation, we obtain the following result.

Theorem 2. *Given a two-player constant-sum game Γ, deciding whether a particular action a is eliminable can be decided in polynomial time.*

Proof. Without loss of generality we may assume that $a \in A_1$. Now consider the algorithm that performs the following steps:

1. Compose a list $(b_1, c_1), \ldots, (b_k, c_k)$ of all actions $b_i \in A_1$ and $c_i \in A_2$ such that c_i backs the elimination of a by b_i.
2. For each i with $1 \leq i \leq k$ arbitrarily eliminate any actions *distinct from* b_i *and* c_i until no more eliminations are possible. Let $\sigma^i = (\sigma^i_1, \ldots, \sigma^i_{m_i})$ denote the resulting valid elimination sequence.
3. If for some i with $1 \leq i \leq k$, action a is eliminated in σ_i, i.e., $a \in \{\sigma^i_1, \ldots, \sigma^i_{m_i}\}$, output "yes", otherwise "no".

Obviously, this algorithm runs in polynomial time. Moreover, soundness follows from Lemma 4. □

5 Win-Lose Games

Conitzer and Sandholm [6] show that subgame reachability and eliminability are NP-complete in *win-lose* games, i.e., games which only allow the outcomes $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$. As both win-lose and constant-sum games generalize single-winner games, it is interesting to compare their results with the ones for constant-sum games in the previous section. We show that Conitzer and Sandholm's results even hold for win-lose games with at most one winner, i.e., for games with $(0, 0)$, $(0, 1)$, $(1, 0)$ as only outcomes.

Theorem 1 established that subgame reachability is NP-complete for games with outcomes in $\{(0, 1), (1, 0)\}$. This obviously implies that this problem is also NP-complete when additionally allowing outcome $(0, 0)$.

It turns out that eliminability is also NP-complete for win-lose games with at most one winner. The proof is a reduction from *3SAT* and involves a modification of the construction used in the proof of Theorem 1.

Theorem 3. *Given a game Γ with outcomes in $\{(0, 0), (0, 1), (1, 0)\}$, deciding whether a particular action is eliminable is NP-complete.*

Conitzer and Sandholm [6] reduced eliminability to solvability in win-lose games to establish the computational intractability of the latter problem. Their construction hinges on the presence of the outcome $(1, 1)$. Our reduction for the more restricted class of games without $(1, 1)$ as an outcome is directly from *3SAT* and exploits the internal structure of the construction used in the proof of Theorem 3.

Theorem 4. *Deciding whether a game Γ with outcomes in $\{(0, 0), (0, 1), (1, 0)\}$ is solvable is NP-complete.*

6 Conclusion

We investigated the computational complexity of iterated weak dominance in two-player constant-sum games. In particular, we showed that deciding whether an action is eliminable is feasible in polynomial time whereas deciding whether a given subgame is reachable is NP-complete. Furthermore, we proved that typical problems associated with iterated dominance in win-lose games *with at most one winner* are NP-complete.

Conitzer and Sandholm [6] have shown that in win-lose games an action is dominated by a mixed strategy if and only if it is dominated by a pure strategy. Thus, although our analysis has been restricted to dominance by pure strategies, it is readily appreciated that all of our results, apart from Theorem 2, immediately extend to dominance by mixed strategies.

A solution concept related to weak dominance is *very weak* dominance, which also allows a player to eliminate one of two actions between which he is completely indifferent. Knuth et al. [9] have shown that in constant-sum games deciding whether an action can be eliminated is P-complete. It is not very hard to see that all problems considered in this paper are tractable when replacing weak dominance with very weak dominance. In a similar spirit, Marx and Swinkels [13] have shown that, in constant-sum games, all subgames that are reachable via iterated *weak* dominance (subject to no more eliminations being possible) are payoff-equivalent in the sense that they are the same up to the addition or removal of identical actions. However, this property does not imply any of our results because it does not discriminate between actions that yield identical payoffs for *some* reachable subgame. The conceptual difference between this paper and the above mentioned work is linked to the question whether one is interested in *action profiles* or *payoff profiles* as "solutions" of a game, or, more generally, whether one champions a prescriptive or a descriptive interpretation of game theory. It may be argued that the computational gap between both concepts is of particular interest in this context.

Acknowledgements. This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR-2312/6-1 and BR 2312/3-2.

References

1. Apt, K.R.: Uniform proofs of order independence for various strategy elimination procedures. Contributions to Theoretical Economics 4(1) (2004)
2. Bernheim, B.: Rationalizable strategic behavior. Econometrica 52(4), 1007–1028 (1984)
3. Brandenburger, A., Friedenberg, A., Keisler, H.J.: Admissibility in games. Econometrica 76(2), 307–352 (2008)
4. Brandt, F., Brill, M., Fischer, F., Harrenstein, P.: Computational aspects of Shapley's saddles. In: Proc. of 8th AAMAS Conference, pp. 209–216 (2009)
5. Brandt, F., Fischer, F., Harrenstein, P., Shoham, Y.: Ranking games. Artificial Intelligence 173(2), 221–239 (2009)
6. Conitzer, V., Sandholm, T.: Complexity of (iterated) dominance. In: Proc. of 6th ACM-EC Conference, pp. 88–97. ACM Press, New York (2005)
7. Ewerhart, C.: Iterated weak dominance in strictly competitive games of perfect information. Journal of Economic Theory 107, 474–482 (2002)
8. Gilboa, I., Kalai, E., Zemel, E.: The complexity of eliminating dominated strategies. Mathematics of Operations Research 18(3), 553–565 (1993)
9. Knuth, D.E., Papadimitriou, C.H., Tsitsiklis, J.N.: A note on strategy elimination in bimatrix games. Operations Research Letters 7, 103–107 (1988)
10. Kohlberg, E., Mertens, J.-F.: On the strategic stability of equilibria. Econometrica 54, 1003–1037 (1986)
11. Koller, D., Megiddo, N.: The complexity of two-person zero-sum games in extensive from. Games and Economic Behavior 4(4), 528–552 (1992)
12. Luce, R.D., Raiffa, H.: Games and Decisions: Introduction and Critical Survey. John Wiley & Sons Inc., Chichester (1957)
13. Marx, L.M., Swinkels, J.M.: Order independence for iterated weak dominance. Games and Economic Behavior 18, 219–245 (1997)
14. Myerson, R.B.: Game Theory: Analysis of Conflict. Harvard University Press, Cambridge (1991)
15. Osborne, M.: An Introduction to Game Theory. Oxford University Press, Oxford (2004)
16. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
17. Pearce, D.: Rationalizable strategic behavior and the problem of perfection. Econometrica 52(4), 1029–1050 (1984)
18. Samuelson, L.: Dominated strategies and common knowledge. Games and Economic Behavior 4, 284–313 (1992)
19. Shimoji, M.: On the equivalence of weak dominance and sequential best response. Games and Economic Behavior 48, 385–402 (2004)
20. Shoham, Y., Leyton-Brown, L.: Multiagent Systems – Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2009)

Swap Bribery

Edith Elkind[1,2], Piotr Faliszewski[3], and Arkadii Slinko[4]

[1] School of ECS, University of Southampton, UK
[2] Division of Mathematical Sciences, Nanyang Technological University, Singapore
[3] Dept. of Computer Science, AGH Univ. of Science and Technology, Kraków, Poland
[4] Dept. of Mathematics, University of Auckland, New Zealand

Abstract. In voting theory, *bribery* is a form of manipulative behavior in which an external actor (the briber) offers to pay the voters to change their votes in order to get her preferred candidate elected. We investigate a model of bribery where the price of each vote depends on the amount of change that the voter is asked to implement. Specifically, in our model the briber can change a voter's preference list by paying for a sequence of swaps of consecutive candidates. Each swap may have a different price; the price of a bribery is the sum of the prices of all swaps that it involves. We prove complexity results for this model, which we call *swap bribery*, for a broad class of voting rules, including variants of approval and k-approval, Borda, Copeland, and maximin.

1 Introduction

There is a range of situations in social choice where an external actor may alter some of the already submitted votes, or the votes that the voters intend to submit. For example, a candidate can attempt to change the voters' preferences by running a campaign, which may be targeted at a particular group of voters. A more extreme (and illegal) version of this strategy involves paying voters to change their votes, or bribing election officials to get access to already submitted ballots in order to modify them. Alternatively, one can assume that the submitted votes can be contaminated with random mistakes, and a central authority should be allowed to correct the votes (preferably, by changing them as little as possible) to reveal the true winner. Indeed, this scenario is, in fact, one of the original motivations behind Dodgson's voting rule. (See also papers [16, 6].)

All of these activities can be interpreted as changing the voters' preferences subject to a budget constraint, and can therefore be studied using the notion of bribery in elections introduced by Faliszewski, Hemaspaandra, and Hemaspaandra [10]. In their model of bribery, we are given an election (i.e., a set of candidates and a list of votes), a preferred candidate p, a price of each vote, and a budget B. We ask if there is a way to pick a group of voters whose total price is at most B so that via changing their votes we can make p a winner.

In the model of Faliszewski, Hemaspaandra, and Hemaspaandra [10] each voter may have a different price, but this price is fixed and does not depend on the nature of the requested change: upon paying a voter, the briber can modify her vote in any way. While there are natural scenarios captured by this model, it fails to express the fact that voters may be more willing to make a small change to their vote (e.g., swap their 2nd

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 299–310, 2009.

and 3rd most favorite candidates) than to change it completely. To account for such settings, Faliszewski [9] proposed a new notion of bribery, which he called *nonuniform bribery*. Under nonuniform bribery, a voter's price may depend on the nature of changes she is asked to implement. A similar notion called *microbribery* was considered in [11]. However, none of these papers considers the standard model of elections, in which votes are preference orders over the set of candidates. Specifically, Faliszewski [9] focused on the so-called utility-based voting, while Faliszewski et al. [11] used the irrational voter model, in which voters' preferences may contain cycles.

The goal of this paper is to study a notion of nonuniform bribery that can be used within the standard model of elections. Our framework, which we call *swap bribery*, is inspired by Dodgson voting rule (see Fellows, Rosamond, and Slinko [12] for a related discussion). We use the name "swap bribery" as it precisely captures the nature of our framework. Specifically, in swap bribery, the briber can ask a voter to perform a sequence of swaps; each swap changes the relative order of two candidates that are currently adjacent in this voter's preference list. For example, if a voter prefers a to b and b to c (we write this as $a \succ b \succ c$), she can be asked to swap a and b, then a and c, then b and c, resulting in the vote $c \succ b \succ a$. Each swap has an associated price, and the total price is simply the sum of the prices of individual swaps. When preferences are viewed as orderings, a swap of adjacent candidates is a natural "atomic" operation on a vote. Moreover, one can transform any vote into any other vote by a sequence of such swaps. Hence, attaching prices to such operations provides a good model for nonuniform bribery in the standard setting.

We also study a special case of swap bribery, which we call *shift bribery*. Under this model of bribery the only allowable swaps are the ones that involve the preferred candidate. Thus, in effect, a shift bribery amounts to asking a voter to move the preferred candidate up by a certain number of positions in her preference order. As argued above, bribery can be used to model a legitimate approach to influencing elections, namely, campaigning: the "briber" simply invests money into trying to convince a particular group of voters that one candidate is better than another. The message and costs of the campaign can vary from one group of voters to another, which is captured by different bribery prices. In this context, shift bribery corresponds to campaigning for the preferred candidate (as opposed to discussing relative merits of other candidates), and is therefore particularly appealing.

After introducing our model of bribery, we proceed to study it from the algorithmic perspective. Our goal here is threefold. First, as argued above, despite its negative connotations, bribery may correspond to perfectly legal and even desirable behavior, and therefore we are interested in developing efficient algorithms that a potential "briber" (that is, a campaign manager) can use. Second, from a more technical perspective, we would like to pinpoint the source of computational hardness in nonuniform bribery. Indeed, when the number of candidates is unbounded, the general bribery of Faliszewski, Hemaspaandra, and Hemaspaandra [10] appears to be hard for all but the simplest voting rules. In contrast, there is a number of polynomial-time algorithms for nonuniform bribery in non-standard models, such as utility-based voting or irrational voters. We would like to know whether these easiness results are tied to the increased flexibility of pricing in nonuniform bribery, or to the increased flexibility of the alternative voter models. The results of this paper, most of which are NP-completeness results, suggest that the latter is true. We are also motivated by the "computational hardness as a barrier against manipulation" line of work, pioneered by Bartholdi, Tovey, and Trick [1]. While it has since been argued that NP-hardness might not provide sufficient protection against dishonest behavior and that more robust notions of hardness are needed (see, e.g., [21, 13, 19, 20]), identifying settings in which bribery is NP-hard is a useful first step towards finding a voting rule that is truly resistant to dishonest behavior.

This paper is organized as follows. After providing the necessary background in Section 2, in Section 3 we formally define our model of bribery, and prove some general results about swap bribery. Section 4 contains a detailed study of bribery in approval voting. In Section 5, we consider other popular voting rules, such as Borda, Copeland, and maximin. We conclude with several directions for further research in Section 6. We omit most of the proofs due to space constraints; these proofs appear in the full version of the paper [7].

2 Preliminaries

Elections. An *election* is a pair $E = (C, V)$, where $C = \{c_1, \dots, c_m\}$ is a set of *candidates* and $V = (v_1, \dots, v_n)$ is a list of *voters*. Each voter v_i is represented via her *preference order* $\succ_i$, which is a strict linear order over the candidates in C (in the context of the possible-winner problem we also allow partial orders). For example, given $C = \{c_1, c_2, c_3\}$ and $V = (v_1, v_2)$, we write $c_2 \succ_2 c_1 \succ_2 c_3$ to denote that the second voter, v_2, prefers c_2 to c_1 to c_3. For any $C' \subseteq C$, by writing C' in a preference order we mean listing all the elements of C' in an arbitrary but fixed order. Similarly, $\overleftarrow{C'}$ means listing members of C' in the reverse of this fixed order.

A *voting rule* $\mathcal{E}$ maps an election $E = (C, V)$ to a set $W \subseteq C$ of *winners*. We assume the nonunique-winner model: all members of $\mathcal{E}(E)$ are considered to be winning. All voting rules considered in this paper are point-based: they assign, via some algorithm, points to candidates, and declare as winners the ones with most points. For an election $E = (C, V)$, we denote by $\mathrm{score}_E(c_i)$ the number of points that a candidate $c_i \in C$ receives in E according to a given voting rule. Sometimes, to disambiguate, we will indicate in the superscript the particular voting rule used. We will provide the definitions of the relevant rules as we discuss them in further sections.

Manipulation, Possible Winners, and Bribery. In this paper we take *manipulation* to mean unweighted constructive coalitional manipulation as defined by Conitzer, Lang, and Sandholm [5]. That is, in $\mathcal{E}$-manipulation we are given an election $E = (C, V)$, a preferred candidate p, and a list of "manipulative" voters V', and we ask if it is possible to set the preferences of voters in V' so that p is an $\mathcal{E}$-winner of $(C, V \cup V')$. In the $\mathcal{E}$-possible-winner problem we are given an election $E = (C, V)$, where the voters' preference are (possibly) *partial*, i.e., are given by partial orders over C, and we ask if it is possible to *complete* the votes so that a given candidate p is an $\mathcal{E}$-winner of the resulting election. It is not hard to see that $\mathcal{E}$-manipulation is a special case of $\mathcal{E}$-possible-winner where some votes are completely specified and some (i.e., those of the manipulative voters) are completely unspecified . The study of possible-winner problems was initiated by Konczak and Lang [15] and then continued by multiple other authors (see, e.g., Walsh's overview paper [17] and the work of Xia and Conitzer [18]). Finally, in $\mathcal{E}$-bribery [10], we are given an election $E = (C, V)$, a preferred candidate p, a list of voters' prices and a nonnegative integer B, and we ask if it is possible to modify votes at a total cost of at most B so that p becomes an $\mathcal{E}$-winner of the resulting election. (In [10] "bribery" refers to the case where all voters have unit prices, and the more general setting described above is called \$bribery.)

Computational Complexity. We assume familiarity with standard notions of computational complexity such as the classes P and NP, NP-completeness, and (polynomial-time) many-one reductions. Many of our hardness proofs rely on reductions from the NP-complete problem EXACT COVER BY 3-SETS (X3C) [14].

Definition 1 ([14]). *An instance* $(\mathcal{B}, \mathcal{S})$ *of* EXACT COVER BY 3-SETS (X3C) *is given by a ground set* $\mathcal{B} = \{b_1, \ldots, b_{3K}\}$*, and a family* $\mathcal{S} = \{S_1, \ldots, S_M\}$ *of subsets of* $\mathcal{B}$*, where* $|S_i| = 3$ *for each* $i = 1, \ldots, M$*. It is a "yes"-instance if there is a subfamily* $\mathcal{S}' \subseteq \mathcal{S}$*,* $|\mathcal{S}'| = K$*, such that for each* $b_i \in \mathcal{B}$ *there is an* $S_j \in \mathcal{S}'$ *such that* $b_i \in S_j$*, and a "no"-instance otherwise.*

3 Swap Bribery

In any reasonable model of nonuniform bribery, one should be able to specify the price for getting a given voter to submit any preference ordering (some of these orderings may be unacceptable to the voter, in which case the corresponding price should be set to $+\infty$). However, in elections with m candidates, there are $m!$ possible votes, so listing the prices of these votes explicitly is not practical. Alternatively, one could specify the bribery prices via an oracle, i.e., via a polynomial-time algorithm that, given a voter i and a preference order $\succ$, outputs the price for getting i to vote according to $\succ$. However, without any restrictions on the oracle, even finding a cheapest way to affect a given vote will require exponentially many queries, and providing appropriate restrictions would be challenging.

Against this background, we will now present a model of bribery that allows for easy specification of bribery prices, and yet is expressive enough to capture many interesting scenarios. Our model is based on the following idea. Intuitively, an atomic operation on a given vote is a swap of two consecutive candidates. Moreover, one can transform any vote into any other vote by a sequence of such steps. It is therefore natural to assume that the price for such transformation is reasonably well approximated by the sum of the prices of individual swaps. We now proceed to formalize this approach.

Let $E = (C, V)$ be an election, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots v_n)$. A *swap-bribery price function* is a mapping $\pi \colon C \times C \to \mathbb{N}$, which for any ordered pair of candidates (c_i, c_j) specifies a number $\pi(c_i, c_j)$, Intuitively, this number is the price of swapping c_i and c_j in a given voter's preference order. More precisely, for a voter v_k with a swap-bribery price function π_k, a *unit swap* is a triple (v_k, c_i, c_j). A unit swap is *admissible* if c_i immediately precedes c_j in v_k's preference order; its price is $\pi_k(c_i, c_j)$. Executing an admissible unit swap (v_k, c_i, c_j) means changing v_k's preference order from $\ldots \succ c_i \succ c_j \succ \ldots$ to $\ldots \succ c_j \succ c_i \succ \ldots$.

Note that we do not allow swapping non-adjacent candidates in a single step (though, of course, such a swap could be simulated by a sequence of swaps of adjacent candidates). Indeed, such a swap would change these candidates' order relative to all candidates that appear between them in the vote.

Definition 2. *For any voting rule $\mathcal{E}$, an instance of $\mathcal{E}$-*swap-bribery *is given by an election $E = (C, V)$ with $C = \{c_1, \dots, c_m\}$, $p = c_1$ and $V = (v_1, \dots, v_n)$, a list of voters' swap-bribery price functions $(\pi_1, \dots, \pi_n)$, and a nonnegative integer B (the budget). We ask if there exists a sequence $(s_1, \dots, s_t)$ of unit swaps such that (1) when executed in order, each unit swap is admissible at the time of its execution, (2) executing $s_1, \dots, s_t$ ensures that p is a winner of the resulting $\mathcal{E}$-election, and (3) the sum of the prices of executing $s_1, \dots, s_t$ is at most B.*

As argued above, swap bribery can be used to transform any vote into any other vote. It is natural to ask if one can efficiently compute an optimal way of doing so. It turns out that the answer to this question is "yes".

Proposition 1. *Given two votes $v_1 = c_{i_1} \succ_1 \dots \succ_1 c_{i_m}$ and $v_2 = c_{j_1} \succ_2 \dots \succ_2 c_{j_m}$, and a swap-bribery price function π, one can compute in polynomial time the cheapest (with respect to π) sequence of swaps converting v_1 into v_2.*

Proof. Set $\mathcal{I}(v_1, v_2) = \{(c_i, c_j) \mid c_i \succ_1 c_j, c_j \succ_2 c_i\}$; we say that a pair of candidates $(c_i, c_j) \in \mathcal{I}(v_1, v_2)$ is *inverted*. Clearly, to obtain v_2 from v_1, it is necessary to swap each inverted pair, so the total cost of an optimal bribery is at least $s = \sum_{(c_i,c_j)\in\mathcal{I}(v_1,v_2)} \pi(c_i, c_j)$. We will now argue that one never needs to swap a pair not in $\mathcal{I}(v_1, v_2)$, or to swap a pair in $\mathcal{I}(v_1, v_2)$ more than once; this implies that the cost of an optimal bribery is exactly s.

Our argument is by induction on the size of $\mathcal{I}(v_1, v_2)$. If $|\mathcal{I}(v_1, v_2)| = 0$, then $v_1 = v_2$ and the statement is obvious. Now, suppose that the statement has been proved for all v_1', v_2' with $|\mathcal{I}(v_1', v_2')| < k$, and consider a pair (v_1, v_2) with $|\mathcal{I}(v_1, v_2)| = k$. We claim that there is a pair of candidates $(c_i, c_j) \in \mathcal{I}(v_1, v_2)$ that is adjacent in v_1. Indeed, suppose otherwise, and let (c_i, c_j) be a pair in $\mathcal{I}(v_1, v_2)$ that is the closest in v_1. By our assumption, there exists at least one $c \in C$ such that $c_i \succ_1 c \succ_1 c_j$, yet $(c_i, c) \notin \mathcal{I}(v_1, v_2)$, $(c, c_j) \notin \mathcal{I}(v_1, v_2)$. Hence, we have $c_i \succ_2 c$, $c \succ_2 c_j$, so by transitivity of $\succ_2$ we conclude $c_i \succ_2 c_j$, a contradiction with $(c_i, c_j) \in \mathcal{I}(v_1, v_2)$. Hence, $\mathcal{I}(v_1, v_2)$ always contains an adjacent pair (c_i, c_j). By swapping c_i and c_j, we obtain a vote v_1' that satisfies $|\mathcal{I}(v_1', v_2)| = k - 1$. Note also that $\mathcal{I}(v_1', v_2) = \mathcal{I}(v_1, v_2) \setminus \{(c_i, c_j)\}$, as the relative order of all other candidates with respect to c_i and c_j did not change. We can now apply our inductive hypothesis. Note that this argument implies a polynomial-time algorithm for transforming v_1 into v_2 in s steps. □

Proposition 1 shows how to optimally convert one vote into another using swaps. We can also compute in polynomial time the cheapest way of transforming a collection of votes into any other collection of votes of the same cardinality.

Proposition 2. *Given a list of votes $V = (v_1, \dots, v_n)$, a corresponding list of price functions $(\pi_1, \dots, \pi_n)$, and a multiset of votes $V' = \{v_1', \dots, v_n'\}$, one can find in polynomial time an optimal swap bribery that transforms V into V'.*

The idea of the proof is to find a minimum-cost perfect matching between V and V', where the cost of each edge (v, v') is given by the price of transforming v into v' via swap bribery.

A voting rule is called *anonymous* if its outcome does not depend on the order of votes in V. Typical voting rules are anonymous. For such rules, Proposition 2 suggests a polynomial-time algorithm for finding an optimal swap bribery in the important special case where the number of candidates is fixed.

Theorem 1. *For any anonymous voting rule with a polynomial-time winner determination procedure, one can compute an optimal swap bribery in polynomial time if the number of candidates is bounded by a constant.*

The idea of the proof is to consider all possible multisets of votes that the briber might request to obtain and apply Proposition 2 to each of them. Observe that when $|C|$ is constant, the number of different multisets of votes is polynomial in $|V|$, but the number of different lists of votes is exponential in $|V|$. This is why Proposition 2 is phrased in terms of multisets of votes rather than lists of votes.

The next result allows us to quickly derive swap-bribery hardness results from possible-winner hardness results.

Theorem 2. *For any voting rule $\mathcal{E}$, $\mathcal{E}$-possible-winner many-one reduces to $\mathcal{E}$-swap-bribery.*

Proof. An instance of the $\mathcal{E}$-possible-winner problem is a pair $\langle (C, V), p\rangle$, where V may contain partial orders and $p \in C$. We will now describe a polynomial-time algorithm that transforms $\langle (C, V), p\rangle$ into an instance of $\mathcal{E}$-swap-bribery in which p can become a winner via swap bribery of cost 0 if and only if the votes in V can be completed in such a way that p is a winner of the resulting election.

Our construction works as follows. First, for each (possibly) partial vote $\succ_k$ in V we compute a complete vote $\succ'_k$ that agrees with $\succ_k$ wherever $\succ_k$ is defined. This can easily be done via, e.g., topological sorting. Next, for each vote $\succ'_k$ we construct a price function π_k as follows. For any pair of candidates $c_i, c_j \in C$, we set $\pi_k(c_i, c_j) = 1$ if $c_i \succ_k c_j$ and $\pi_k(c_i, c_j) = 0$ otherwise. We output an instance of swap bribery with budget 0, preferred candidate p and an election E' which is identical to E except that each vote $\succ_k$ is replaced by vote $\succ'_k$ associated with price function π_k.

Clearly, this reduction works in polynomial time. To prove its correctness, fix an index k and consider a vote $\succ'_k$ and an arbitrary vote $\succ''_k$. We claim that $\succ'_k$ can be transformed into $\succ''_k$ via a swap bribery of cost 0 (with respect to π_k) if and only if $\succ''_k$ agrees with $\succ_k$ on all pairs of candidates comparable under $\succ_k$. Indeed, as shown in the proof of Proposition 1, the optimal swap bribery that transforms $\succ'_k$ into $\succ''_k$ swaps each pair of candidates c_i, c_j such that $c_i \succ'_k c_j$ and $c_j \succ''_k c_i$ exactly once. Clearly, the cost of these swaps is 0 if and only if $\succ''_k$ agrees with $\succ_k$ on all pairs of candidates comparable under $\succ_k$. Consequently, the votes in E can be completed so as to make p a winner if and only if there is a swap bribery of cost 0 that makes p a winner in E'. $\square$

Since $\mathcal{E}$-manipulation is a special case of $\mathcal{E}$-possible-winner, as a corollary we immediately obtain that $\mathcal{E}$-manipulation many-one reduces to $\mathcal{E}$-swap-bribery.

Shift bribery. In some settings, the briber may be unable to ask voters to make a swap that does not involve the preferred candidate. For example, in an election campaign investing money to support another candidate may be viewed as unethical. In such cases, the only action available to the briber is to ask a voter to move the preferred candidate up in her preference order. We will refer to this type of bribery as *shift bribery*.

Fix an election $E = (C, V)$ with $C = \{c_1, \dots, c_m\}$, $p = c_1$, and a voter $v \in V$ with a preference order $\succ$. Suppose that p appears in the jth position in $\succ$. We say that a mapping $\rho : \mathbb{N} \to \mathbb{N}$ is a *shift-bribery price function* for v if it satisfies (1) $\rho(0) = 0$; (2) $\rho(i) \leq \rho(i')$ for $i < i' < j$; and (3) $\rho(i) = +\infty$ for $i \geq j$. We interpret $\rho(i)$ as the price of moving p up by i positions in $\succ$.

Definition 3. *For any voting rule $\mathcal{E}$, an instance of $\mathcal{E}$-*shift-bribery *is given by an election $E = (C, V)$ with $C = \{c_1, \dots, c_m\}$, $p = c_1$ and $V = (v_1, \dots, v_n)$, a list of voters' shift-bribery price functions $(\rho_1, \dots, \rho_n)$, and a nonnegative integer B (the budget). We ask if there is a sequence $(k_1, \dots, k_n)$ of nonnegative integers such that $\sum_{i=1}^{n} \rho_i(k_i) \leq B$ and bribing each voter v_i to shift p up by k_i places ensures that p is a $\mathcal{E}$-winner of the resulting election.*

It is not hard to see that $\mathcal{E}$-shift-bribery is a special case of $\mathcal{E}$-swap-bribery.

Proposition 3. *For any voting rule $\mathcal{E}$, any election $E = (C, V)$ with $C = \{c_1, \dots, c_m\}$, $p = c_1$ and $V = (v_1, \dots, v_n)$, and any list $(\rho_1, \dots, \rho_n)$ of shift-bribery price functions for V, we can efficiently construct a list $(\pi_1, \dots, \pi_n)$ of swap-bribery price functions for V so that the problem of $\mathcal{E}$-shift-bribery with respect to $(\rho_1, \dots, \rho_n)$ is equivalent to the problem of $\mathcal{E}$-swap bribery with respect to $(\pi_1, \dots, \pi_n)$.*

Proof. The general idea of the proof is as follows. We set the budget in the swap bribery problem to be the same as in the input shift bribery problem. To construct a swap-bribery price function π_i for a voter v_i, we renumber the candidates in C so that $c_1 = p$ and v_i's preference order is $c_k \succ_i c_{k-1} \succ_i \cdots \succ_i c_2 \succ_i p \succ_i \cdots$. Now set

$$\pi_i(x, y) = \begin{cases} \rho_i(1) & \text{if } x = p \text{ and } y = c_2 \\ \rho_i(\ell - 1) - \rho_i(\ell - 2) & \text{if } x = p \text{ and } y = c_\ell, \ell = 3, \dots, k \\ +\infty & \text{in all other cases.} \end{cases}$$

A simple inductive proof shows that setting all π_i in this way proves the theorem. □

The analog of Theorem 2 does not seem to hold for shift bribery. Hence, unlike in the case of swap bribery, it is of interest to explore the complexity of shift bribery even when the corresponding possible-winner problem is known to be hard. Another natural question in this context is whether there are voting rules for which shift bribery is strictly easier than swap bribery. As our subsequent results show, the answer to this question is "yes" (assuming P $\neq$ NP).

4 Case Study: Approval Voting

In this section we investigate the complexity of swap bribery in k-approval voting. The family of k-approval voting rules (for various values of k) is a simple but interesting class of voting rules, including such well-known rules as plurality and veto. In

k-approval, a voter assigns a point to each of the top k candidates on her preference list. Thus, 1-approval is simply the *plurality* rule and, for $|C| = m$, $(m-1)$-approval is the *veto* rule, where, in effect, each voter votes against her least desirable candidate. Our first result is that swap bribery is easy for plurality and veto but hard for almost all variants of k-approval with fixed k.

Theorem 3. *Swap bribery is in* P *for plurality (i.e.,* 1*-approval) and veto (i.e.,* $(m-1)$*-approval). However, for each fixed* k *such that* $k \geq 3$*, swap bribery for* k*-approval is* NP*-complete, even if all swaps have costs in the set* $\{0, 1, 2\}$.

We omit the proof of this theorem due to space constraints. Note that Theorem 3 does not say anything about the complexity of swap bribery for 2-approval. Very recently, Betzler and Dorn [2] have shown that for 2-approval the possible winner problem is NP-hard, and thus by Theorem 2 swap bribery for 2-approval is NP-hard as well.
In contrast to Theorem 3, shift bribery for k-approval is easy for all values of k. Thus, shift bribery can indeed be easier than swap bribery.

Theorem 4. *Shift bribery for* k*-approval is in* P *for any* $k < m$.

The proof relies on the fact that under shift bribery, the only reasonable way to bribe a given voter is to ask him to approve of p at the lowest possible cost.

Now, the NP-hardness proof in Theorem 3 assumes that both the number of candidates and the number of voters are parts of the input (i.e., are not bounded by any fixed constant). We have seen that the first requirement is necessary: by Theorem 1, swap bribery becomes easy if the number of candidates is constant. It is therefore natural to ask if the number of voters plays a similar role. It turns out that if k is bounded by a constant, swap bribery is indeed easy for each fixed number of voters.

Theorem 5. *For each fixed* k*, swap bribery for* k*-approval is in* P *if the number of voters is bounded by a constant.*

Proof. Consider an election $E = (C, V)$ with $C = \{c_1, \dots, c_m\}$, $V = (v_1, \dots, v_n)$, a preferred candidate $p \in C$, a budget B. and a list of price functions $(\pi_1, \dots, \pi_n)$. Let $C_1, \dots, C_T$ be the list of all k-element subsets of C; note that $T = \binom{m}{k} = \mathrm{poly}(m)$. For a given vote v, we can compute the cost of moving the candidates from a given k-element subset C_t into top k positions in v. Indeed, suppose that $C_t = \{c_{i_1}, \dots, c_{i_k}\}$, and c_{i_1} is the first of these candidates to appear in v, c_{i_2} is second, etc. Then this cost is simply the cost of moving c_{i_1} into the top position by successively swapping it with all candidates that are above him, followed by moving c_{i_2} into the second position, etc. To see why this naive algorithm is optimal, note that it only swaps pairs that are inverted in the sense of Proposition 1, i.e., ones that have to be swapped anyway.

We can now go over all lists of the form $(C_{i_1}, \dots, C_{i_n})$, $i_j \in \{1, \dots, T\}$ for $j = 1, \dots, n$, and for each such list compute the cost of the optimal bribery that for $j = 1, \dots, n$ transforms the jth input vote into a vote that lists the candidates in C_{i_j} in the top k positions. There are at most $\binom{m}{k}^n = \mathrm{poly}(m)$ such lists; we accept if at least one of them costs at most B and bribing the voters to implement it ensures p's victory. □

On the other hand, when k is unbounded, swap bribery becomes difficult even if there is just one voter. To prove this result, we reduce from the NP-complete problem BALANCED BICLIQUE (BB) (see [14]).

Definition 4 ([14]). *An instance of* BB *is given by a bipartite graph* $G = (U, W, E)$, *where* $|U| = |W| = N$ *and* $E \subseteq U \times W$, *and a natural number* $K \leq N$. *It is a "yes"-instance if there are sets* $U' \subseteq U$ *and* $W' \subseteq W$ *such that* $|U'| = |W'| = K$ *and for all* $u \in U'$, $w \in W'$ *we have* $(u, w) \in E$, *and a "no"-instance otherwise.*

Intuitively, the reason why swap bribery for k-approval is difficult for large values of k is that it may be beneficial for the briber to move around some candidates other than p, as this may enable him to promote p via swaps of lower cost.

Theorem 6. *When k is a part of the input, swap bribery for k-approval is* NP*-complete even for a single voter.*

Proof. It is easy to see that our problem is in NP. We focus on the NP-hardness proof. We give a reduction from BB (see Definition 4 above). Suppose that we are given an instance of BB with $U = \{u_1, \dots, u_N\}$, $W = \{w_1, \dots, w_N\}$. Our election will have $2N + 1$ candidates $u_1, \dots, u_N, w_1, \dots, w_N, p$, where p is the preferred candidate, and a single voter v with preference ordering $U \succ W \succ p$. The price function is given by $\pi(u_i, u_j) = 0$, $\pi(w_i, w_j) = 0$ for all $i, j = 1, \dots, N$, $\pi(w_i, p) = 1$, $\pi(u_i, p) = 0$ for all $i = 1, \dots, N$, $\pi(u_i, w_j) = 0$ if $(u_i, w_j) \in E$ and $\pi(u_i, w_j) = N - K + 1$ otherwise. Finally, we set $k = N + 1$ and $B = N - K$.

Suppose that we have a "yes"-instance of BB, and let (U', W') be the corresponding witness. Then we can first reorder U and W for free so that $U \setminus U' \succ U'$, $W' \succ W \setminus W'$, then swap U' and W' (which is free, since (U', W') is a biclique in G), and, finally, move p past $W \setminus W'$ and U', paying $|W \setminus W'| = N - K = B$.

Conversely, suppose that there is a successful bribery for v. Let U' be the set of candidates from U that end up below p, and let W' be the set of candidates from W that end up above p after the bribery. Observe that this means that we had to swap each pair $(u, w) \in U' \times W'$, and hence $(u, w) \in E$ for all $(u, w) \in U' \times W'$, as otherwise we would have exceeded our budget. We had to pay 1 for swapping p with each of the candidates in $W \setminus W'$, so $|W \setminus W'| \leq N - K$ and hence $W' \geq K$. On the other hand, p ended up among the top $N + 1$ candidates, so $|W'| + |U \setminus U'| \leq N$, and hence $|U'| \geq K$. Pick $U'' \subseteq U'$, $W'' \subseteq W'$ so that $|U''| = |W''| = K$. The pair (U'', W'') is a balanced biclique of the required size in G because we have started with a successful bribery. □

Bribery in SP-AV. Nonuniform bribery for approval voting has already been studied thoroughly [10, 9]. Recently, Brams and Sanver [3] introduced a variant of approval voting called SP-AV, whose computational study was initiated by Erdélyi, Nowak, and Rothe [8]. In the full version of this paper [7] we discuss swap bribery for SP-AV.

5 Further Voting Rules and Shift Bribery

In this section we consider voting rules other than approval, starting with Borda. In a Borda election with m candidates, the number of points assigned by a voter v to a candidate c equals the number of candidates that v ranks below c. The possible winner problem for Borda is NP-complete [18] and thus Theorem 2 implies that swap bribery for Borda is NP-complete. Thus, we will now focus on Borda-shift bribery.

Perhaps unsurprisingly, shift bribery for Borda turns out to be computationally hard.

Theorem 7. *Shift bribery for Borda is* NP*-complete.*

However, there exists a 2-approximation algorithm for Borda-shift bribery.

Theorem 8. *There exists a polynomial time algorithm that, given an instance* $I = (C, V, p, (\rho_1, \ldots, \rho_n), B)$ *of shift bribery, outputs a sequence of shifts that makes* p *a Borda winner, and whose cost is at most twice the cost of an optimal Borda-shift bribery for* I.

Proof. Fix an instance I of Borda-shift bribery. Suppose that the optimal shift bribery in I has cost c and moves p up by k positions in total. It is easy to see that *any* bribery in I that shifts p up by at least $2k$ positions makes p a winner. Indeed, in the optimal solution shifting p up by k positions increases p's score by k and decreases every other candidate's score by at most k. Thus, altogether the advantage that p has over any other candidate increases by at most $2k$. We obtain the same effect by shifting p up by $2k$ positions.

Suppose that we know k. Then we can use dynamic programming to compute a minimum-cost bribery that shifts p up by k positions as follows. For each $i = 1, \ldots, n$ and $k' = 1, \ldots, k$, let $f(i, k')$ be the cost of a minimum-cost shift bribery that moves p up by k' positions in the preferences of the first i voters. We have $f(1, k') = \rho_i(k')$ for $k' \leq m - k_1$, where k_1 is the position of p in the first vote, and $f(1, k') = +\infty$ for $k' > m - k_1$. Further, we have $f(i + 1, k') = \min\{f(i, k' - k'') + \rho_{i+1}(k'') \mid k'' = 1, \ldots, m - k_{i+1}\}$, where k_{i+1} is the position of p in the $(i + 1)$st vote. Denote the resulting bribery by $\mathcal{B}$. Obviously, the cost of $\mathcal{B}$ is given by $f(n, k)$, and one can compute $\mathcal{B}$ itself using standard techniques. Observe that the cost of $\mathcal{B}$ is at most c.

The bribery $\mathcal{B}$ includes some j shifts, $j \leq k$, that also appear in the optimal solution. Suppose that we know the value of j, and imagine that we first execute these j shifts. After doing so, we get an instance I' that still allows the remaining $k - j$ shifts of the optimal solution. Thus, given I', one can find $k - j$ shifts that ensure p's victory and so, by the observation in the previous paragraph, any $2(k - j)$ shifts from I' suffice to make p a winner. Let I'' be the instance obtained after executing $\mathcal{B}$. Clearly, one can transform I' into I'' using $k - j$ shifts. Therefore, in I'' any bribery that shifts p by $k - j$ positions makes p a winner. Thus, after executing $\mathcal{B}$, we pick the cheapest bribery $\mathcal{B}'$ that shifts p up by $k - j$ positions. These $k - j$ shifts cost at most c, because there are the $k - j$ unused shifts from the optimal solution, whose cost is at most c. As a result, we ensure p's victory via $2k - j$ shifts, and pay at most $2c$.

Now, the algorithm above assumes that we know k and j. When solving an arbitrary instance, we do not know them, but we can try all combinations. □

There is an interesting connection between shift bribery for Borda and computing the Dodgson score of a candidate. We point the reader to the full version of this paper [7] for a discussion comparing the algorithm described above and one of the algorithms of [4] for the Dodgson score.

We now turn to elections defined via considering majority contests between pairs of candidates. Specifically, we consider maximin and Copeland$^\alpha$, where α is a rational number, $0 \leq \alpha \leq 1$. These voting rules are formally defined as follows. Fix an election $E = (C, V)$ where $C = \{c_1, \dots c_m\}$ and $V = (v_1, \dots v_n)$. and define

$$N_E(c_i, c_j) = |\{v_k \mid c_i \succ_k c_j\}|.$$

Let α be a rational number such that $0 \leq \alpha \leq 1$. Then the Copeland$^\alpha$ score of a candidate c_i, which we denote by $\mathrm{score}_E^\alpha(c_i)$, is defined as

$$\mathrm{score}_E^\alpha(c_i) = |\{c_j \mid N_E(c_i, c_j) > N_E(c_j, c_i)\}| + \alpha|\{c_j \mid N_E(c_i, c_j) = N_E(c_j, c_i)\}|.$$

The maximin score of a candidate c_i, which we denote by $\mathrm{score}_E^m(c_i)$, is defined as $\mathrm{score}_E^m(c_i) = \min_{i \neq j} N_E(c_i, c_j)$.

Theorem 9. *Shift bribery is* NP*-complete for maximin and, for each rational* α *between* 0 *and* 1*, for Copeland*$^\alpha$*.*

It is interesting to compare the results of this section with those of [11], where it is shown that for irrational voters microbribery for Copeland^0 and for Copeland^1 is in P. In fact, we can show that microbribery for the case of irrational voters is in P also for Borda and maximin (though we omit these results due to limited space and our focus on rational voters). This is a further (meta)-argument that perhaps the main source of hardness in many voting problems stems from having to deal with preference orders rather than the properties of particular voting rules.

6 Conclusions

We introduced two notions of nonuniform bribery—swap bribery and shift bribery—for the standard model of elections, and analyzed their complexity for several well-known voting rules such as plurality, k-approval, Borda, Copeland, and maximin. It turns out that, in sharp contrast to the easiness results for microbribery [11] and nonuniform bribery in utility-based systems [9], swap bribery is NP-hard for many of these rules. This is quite surprising as our swap bribery is essentially the microbribery model adapted to the rational-voter setting.

Our work leads to several open problems. First, it would be useful to identify natural special cases of our setting for which one can find an optimal swap bribery in polynomial time. Another way to tackle computational hardness is by constructing efficient approximation algorithms for swap bribery and shift bribery; Theorem 8 makes the first step in this direction. Designing approximation algorithms for shift bribery under other voting rules as well as for swap bribery is an interesting topic for future research.

Acknowledgements. We would like to thank the anonymous referees for very helpful comments and suggestions. We are also grateful to Jörg Rothe for hosting a visit of the first two authors to Heinrich-Heine-Universität, Düsseldorf. Piotr Faliszewski is supported by AGH University of Science and Technology grant no. 11.11.120.777. Edith Elkind is supported by EPSRC grant GR/T10664/01, ESRC grant ES/F035845/1, and by NRF Research Fellowship.

References

[1] Bartholdi III, J., Tovey, C., Trick, M.: The computational difficulty of manipulating an election. Social Choice and Welfare 6(3), 227–241 (1989)
[2] Betzler, N., Dorn, B.: Towards a dichotomy of finding possible winners in elections based on scoring rules. In: Královič, R., Niwiński, D. (eds.) MFCS 2009. LNCS, vol. 5734, pp. 124–136. Springer, Heidelberg (2009)
[3] Brams, S., Sanver, R.: Critical strategies under approval voting: Who gets ruled in and ruled out. Electoral Studies 25(2), 287–305 (2006)
[4] Caragiannis, I., Covey, J., Feldman, M., Homan, C., Kaklamanis, C., Karanikolas, N., Procaccia, A., Rosenschein, J.: On the approximability of Dodgson and Young elections. In: Proceedings of SODA-2009 (2009)
[5] Conitzer, V., Sandholm, T., Lang, J.: When are elections with few candidates hard to manipulate? Journal of the ACM 54(3) (2007)
[6] Elkind, E., Faliszewski, P., Slinko, A.: On distance rationalizability of some voting rules. In: Proceedings of TARK-2009 (2009)
[7] Elkind, E., Faliszewski, P., Slinko, A.: Swap bribery. Technical Report arXiv:0905.3885 [cs.GT], arXiv.org (May 2009)
[8] Erdélyi, G., Nowak, M., Rothe, J.: Sincere-strategy preference-based approval voting broadly resists control. In: Ochmański, E., Tyszkiewicz, J. (eds.) MFCS 2008. LNCS, vol. 5162, pp. 311–322. Springer, Heidelberg (2008)
[9] Faliszewski, P.: Nonuniform bribery (short paper). In: Proceedings of AAMAS-2008 (2008)
[10] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: The complexity of bribery in elections. In: Proceedings of AAAI-2006 (2006)
[11] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Llull and Copeland voting computationally resist bribery and constructive control. Journal of Artificial Intelligence Research (2009)
[12] Fellows, M., Rosamond, F., Slinko, A.: Sensing God's will is fixed parameter tractable. Technical Report N.561, Department of Mathematics. The University of Auckland (July 2008)
[13] Friedgut, E., Kalai, G., Nisan, N.: Elections can be manipulated often. In: Proceedings of FOCS-2008 (2008)
[14] Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
[15] Konczak, K., Lang, J.: Voting procedures with incomplete preferences. In: Proceedins of the Multidisciplinary IJCAI-2005 Worshop on Advances in Preference Handling, July/August 2005, pp. 124–129 (2005)
[16] Meskanen, T., Nurmi, H.: Closeness counts in social choice. In: Braham, M., Steffen, F. (eds.) Power, Freedom, and Voting. Springer, Heidelberg (2008)
[17] Walsh, T.: Uncertainty in preference elicitation and aggregation. In: Proceedings of AAAI-2007 (2007)
[18] Xia, L., Conitzer, V.: Determining possible and necessary winners under common voting rules given partial orders. In: Proceedings of AAAI-2008 (2008)
[19] Xia, L., Conitzer, V.: Generalized scoring rules and the frequency of coalitional manipulability. In: Proceedings of EC-2008 (2008)
[20] Xia, L., Conitzer, V.: A sufficient condition for voting rules to be frequently manipulable. In: Proceedings of EC-2008 (2008)
[21] Zuckerman, M., Procaccia, A., Rosenschein, J.: Algorithms for the coalitional manipulation problem. Artificial Intelligence 173(2), 392–412 (2008)

Performances of One-Round Walks in Linear Congestion Games

Vittorio Bilò[1], Angelo Fanelli[2], Michele Flammini[3], and Luca Moscardelli[4]

[1] Dipartimento di Matematica "Ennio De Giorgi" - Università del Salento
Provinciale Lecce-Arnesano P.O. Box 193, 73100 Lecce, Italy
vittorio.bilo@unisalento.it
[2] Department of Computer Science - RWTH Aachen University, Germany
fanelli@cs.rwth-aachen.de
[3] Dipartimento di Informatica - Università di L'Aquila
Via Vetoio, Coppito 67100 L'Aquila, Italy
flammini@di.univaq.it
[4] Dipartimento di Scienze - Università di Chieti-Pescara
Viale Pindaro 42, 65127 Pescara, Italy
moscardelli@sci.unich.it

Abstract. We investigate the approximation ratio of the solutions achieved after a one-round walk in linear congestion games. We consider the social functions Sum, defined as the sum of the players' costs, and Max, defined as the maximum cost per player, as a measure of the quality of a given solution. For the social function Sum and one-round walks starting from the empty strategy profile, we close the gap between the upper bound of $2+\sqrt{5} \approx 4.24$ given in [8] and the lower bound of 4 derived in [4] by providing a matching lower bound whose construction and analysis require non-trivial arguments. For the social function Max, for which, to the best of our knowledge, no results were known prior to this work, we show an approximation ratio of $\Theta(\sqrt[4]{n^3})$ (resp. $\Theta(n\sqrt{n})$), where n is the number of players, for one-round walks starting from the empty (resp. an arbitrary) strategy profile.

1 Introduction

In congestion games [19] there is a set of n players sharing a set E of resources. Each player can choose among certain subsets of resources. The congestion of a resource $e \in E$ is defined as the number of players using e in a given strategy profile. Each resource has an associated latency function which only depends on its congestion. The cost of a player in a given strategy profile is then defined as the sum of the latencies of all the resources she is using. The cases in which all latency functions are linear (resp. exponential) in the congestion are called linear (resp. exponential) congestion games, while those in which the strategies available to all players are made of a single resource are called singleton congestion games.

Congestion games constitute, perhaps, the most studied class of games in Algorithmic Game Theory for at least two reasons: first, they can model the

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 311–322, 2009.

non-cooperative version of several basic computational problems, such as load balancing, network design with fair cost sharing, routing, facility location, and, secondly, they possess several useful properties suggesting and easying the study of different aspects of selfish and non-cooperative behavior. It is well known, in fact, that congestion games are isomorphic to exact potential games [17], which means that they not only always admit pure Nash equilibria, but that any better-response dynamic always converges to one such an equilibrium independently from the starting strategy profile (finite improvement path property). Moreover, among these games, the subclass of linear congestion games occupies a particular role, since, together with the exponential one, it is the only subclass of congestion games whose weighted version still possesses the finite improvement path property, see [13,14,18]. The generalization of congestion games with weighted players, in fact, may not admit pure Nash equilibria even when the latency functions are monotone in the congestion [15].

In non-cooperative games possessing the finite improvement path property, a natural question is that of evaluating the worst-case number of moves needed to reach a pure Nash equilibrium starting from a given or an arbitrary strategy profile. Also, one may ask about the price of anarchy yielded by the subset of pure Nash equilibria which can be reached starting from a certain strategy profile. Pushing these issues even further, since for the majority of games an exponential number of moves may be needed in order to reach a pure Nash equilibrium, it becomes natural to ask whether a polynomial number of moves suffices to reach strategy profiles whose performances are sufficiently close to those of pure Nash equilibria. In their seminal paper, Mirrokni and Vetta [16] introduced the notions of covering walk and k-round walk in order to model significant sequences of best-response dynamics. A covering walk is a sequence of best-response dynamics in which each player performs at least a move. A one-round walk is a covering walk in which each player performs exactly a move, while a generic k-round walk is the concatenation of k one-round walks. In such a setting it is quite natural to assume the empty strategy profile as the most reasonable starting point for any type of walk.

Our Contribution. In this paper we investigate the approximation ratio of the solutions achieved after a one-round walk in linear congestion games. This study is motivated by the fact that convergence to pure Nash equilibria in these games may require a number of moves which can be exponential in n [1]. We consider the social functions Sum, defined as the sum of the players' costs, and Max, defined as the maximum cost per player, as a measure of the quality of a given strategy profile. For the social function Sum and one-round walks starting from the empty strategy profile, we close the existing gap between the upper bound of $2 + \sqrt{5} \approx 4.24$ given in [8] and the lower bound of 4 derived in [4] by providing a family of instances yielding lower bounds approaching $2 + \sqrt{5}$ as the number of players goes to infinity. The construction and the analysis of these instances require non-trivial arguments. For the social function Max, which, to the best of our knowledge, was not considered before in this setting, we provide asymptotically matching upper and lower bounds which show a $\Theta(\sqrt[4]{n^3})$-approximation and a

$\Theta(n\sqrt{n})$-approximation for one-round walks starting from the empty strategy profile and an arbitrary strategy profile, respectively.

Related Works. Christodoulou and Koutsoupias showed in [7] that the price of anarchy of pure Nash equilibria in linear congestion games is 2.5 for the social function Sum and $\Theta(\sqrt{n})$ for the social function Max. For the social function Sum, Awerbuch et al. [2] gave an exact bound of 2.618 on the price of anarchy of either pure and mixed Nash equilibria in weighted linear congestion games. Again, for the social function Sum, the 2.5 and 2.618 bounds were extended to correlated equilibria by Christodoulou and Koutsoupias in [6], where they also proved that the price of stability of pure Nash equilibria lies between $1+\sqrt{3}/3 \approx 1.577$ and 1.6. Caragiannis et al. [4] closed this gap by lowering the upper bound to exactly $1+\sqrt{3}/3$.

Mirrokni and Vetta [16] initiated the study of convergence of Nash dynamics to approximate solutions by focusing on the class of valid-utility games [21] and, in particular, on two special subclasses: basic-utility games and market sharing games. Such an investigation is particularly relevant for congestion games, since Fabrikant et al. [10] have proven that determining a Nash equilibrium is a PLS-complete problem [9], even when all the players have the same strategy set. Ackermann et al. [1] showed that PLS-completeness carries over also to the case of linear latency functions. As a consequence, a number of moves exponential in the number of players may be required to reach an equilibrium.

However, for the special case of symmetric congestion games on series-parallel networks, Fotakis at al. [13] showed that a one-round walk starting from the empty configuration leads to a pure Nash equilibrium, while Fotakis [12] showed that for symmetric congestion games on extension-parallel networks, a pure Nash equilibrium is reached even when starting from an arbitrary configuration.

Christodoulou et al. [8] initiated the study of the performances of one-round walks in linear congestion games. For the social function Sum, they proved that the approximation ratio of the solutions achieved after a one-round walk is $\Theta(n)$ when starting from an arbitrary strategy profile, while it is at most $2+\sqrt{5} \approx 4.24$ when starting from the empty one. They also provided a linear congestion game for which the approximation ratio of the solutions achieved after a k-round walk is $\Omega(\sqrt[2^{O(k)}]{n}/k)$. Such result has been improved by Fanelli et al. [11] by showing a lower bound of $\Omega(\sqrt[2^k]{n}/k)$ and an asymptotically matching upper bound, thus proving that a $\Theta(\log \log n)$-round walk is necessary and sufficient to achieve a nearly optimal strategy profile. The importance of the results in [11] is emphasized by the existence of congestion games with $\Omega(2^n)$-round walks from an arbitrary state to a Nash equilibrium and by the recent negative result by Awerbuch et al. [3]. They construct a linear congestion game such that an $\Omega(\frac{\sqrt{n}}{\log n})$-approximate solution is achieved after a k-covering walk, where k is exponential in the number of players and there are players performing a polynomial number of moves in each covering walk.

Finally, results about the performances of ϵ-moves in congestion games can be found in [3,5,20].

Paper Organization. Next section contains the necessary definitions and notation. In Sections 3 and 4 we present our results for the social functions Sum and Max, respectively. Finally, in the last section, we deal with open problems and future research. Due to space limitations, many details have been removed and will appear in the full version.

2 Definitions and Preliminaries

(Non-cooperative) Strategic Games. A *strategic game* is defined by a tuple $\mathcal{G} = (N, (\Sigma_i)_{i\in N}, (c_i)_{i\in N})$, where $N = \{1, 2, \ldots, n\}$ denotes the set of n *players* (or *agents*), Σ_i a set of (*pure*) *strategies* for player i and $c_i : \times_{i\in N}\Sigma_i \mapsto \mathbb{R}_{\geq 0}$ is the *cost function* for player i.

Let $\Sigma = \times_{i\in N}\Sigma_i$ be the *strategy profile set* or *state set* of the game and $S = (s_1, s_2, \ldots, s_n) \in \Sigma$ be a generic *strategy profile* or *state* (or *solution*) in which each player i chooses strategy $s_i \in \Sigma_i$. $\emptyset_i$ corresponds to an *empty* (or *null*) *strategy* for player i, and $\emptyset$ to the *empty strategy profile*, i.e., $\emptyset = (\emptyset_1, \emptyset_2, \ldots, \emptyset_n)$. Whereas we call *arbitrary*, any strategy profile in which every player choose a non-null strategy. Given the strategy profile $S = (s_1, s_2, \ldots, s_n)$ and a strategy $s'_i \in \Sigma_i$, let $(S \oplus s'_i) = (s_1, s_2, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_n)$ be the strategy profile obtained from S if player i changes her strategy from s_i to s'_i.

In a *non-cooperative* strategic game we assume that each player acts selfishly and aims at choosing the strategy minimizing her cost, given the strategic choices of the other players. For a strategy profile $S = (s_1, s_2, \ldots, s_n)$, an *improving move* of player i is a strategy $s'_i \in \Sigma_i$ such that $c_i(S \oplus s'_i) < c_i(S)$. Furthermore, a *best response* (*move*) of player i in S is a strategy $s^*_i \in \Sigma_i$ yielding the minimum possible cost, given the strategic choices of the other players, i.e., $c_i(S \oplus s^*_i) \leq c_i(S \oplus s'_i)$ for any other strategy $s'_i \in \Sigma_i$. Notice that a best response move corresponding to the strategy currently played in S by the involved player is not necessarily an improving move.

Best Response Nash Dynamics Graphs. The *best response Nash dynamics graph* associated to a non-cooperative strategic game $\mathcal{G}$ is a directed graph $\mathcal{B} = (V, A)$ where each vertex in V corresponds to a strategy profile and there is an edge $(S, S') \in A$ with label i, where $S' = S \oplus s'_i$ and $s'_i \in \Sigma_i$, if and only if both the following conditions are met: (I) s'_i is a best response move of i in S; (II) if $S \neq S'$, s'_i is also an improvement move of i in S. Any sink in $\mathcal{B}$ corresponds to a *pure Nash equilibrium*. Observe that $\mathcal{B}$ may contain cycles.

A *best response walk* in $\mathcal{B}$ is a directed walk $W = (S^0, S^1, \ldots, S^k)$. We denote by $\pi_W(i)$ the label of the edge (S^i, S^{i+1}). S^0 is said the *initial* state of W and S^k its *final* state. In this work we consider the following type of best response walks.

- One-round walk (or round) : it is a best response walk in $\mathcal{B}$ with exactly one edge with label i, for each player $i \in N$. We denote the round by $\big(S^0, S^1, \ldots, S^n\big)$. Let $S^0 = (s_1, s_2, \ldots, s_n)$. We assume that the player moving in state S^{i-1} is player i, which changes her strategy from s_i to s'_i, that is $S^i = S^{i-1} \oplus s'_i = (s'_1, s'_2, \ldots, s'_i, s_{i+1}, \ldots, s_n)$.

Congestion Games. A *congestion game* $\mathcal{C} = (N, E, (\Sigma_i)_{i\in N}, (f_e)_{e\in E}, (c_i)_{i\in N})$ is a non-cooperative strategic game in which there is a set E of m *resources* (or *facilities*) to be shared among the players in N. Any (pure) strategy $s_i \in \Sigma_i$ for player i is a subset of resources, i.e., $\Sigma_i \subseteq 2^E$. Given a strategy profile $S = (s_1, s_2, \ldots, s_n)$ and a resource e, the number of players using e in S, called the *congestion* on e, is denoted by $n_e(S) = |\{i \in N \mid e \in s_i\}|$. A *latency function* $f_e : \mathbb{N} \mapsto \mathbb{R}_{\geq 0}$ associates to resource e a cost (or latency) depending on the number of players currently using e. The *cost* (or *latency*) of player i for the pure strategy s_i, depends on the congestion of each resource in s_i and is given by $c_i(S) = \sum_{e\in s_i} f_e(n_e(S))$. This work is concerned only with *linear latency* functions, i.e., the case in which $f_e(x) = a_e \cdot x + b_e$ with $a_e, b_e \in \mathbb{R}_{\geq 0}$. Moreover, we assume that the *social cost* of S can be either the sum (or average) of the players' costs, i.e., $\textsc{Sum}(S) = \sum_{i\in N} c_i(S)$, or the maximum cost of a player, i.e., $\textsc{Max}(S) = max_{i\in N} c_i(S)$. We denote the cost of the *optimal* strategy profile, i.e., the strategy profile minimizing the social cost, by $\textsc{Opt}_{sum}$ and $\textsc{Opt}_{max}$ respectively.

3 Social Function Sum

Christodoulou et al. [8] proved that for any linear congestion game and one-round walk $(S^0, S^1, \ldots, S^n)$ with $S^0 = \emptyset$, it holds $\textsc{Sum}(S^n) \leq (2+\sqrt{5})\textsc{Opt}_{sum} \approx 4.24\textsc{Opt}_{sum}$. Surprisingly enough, the best known lower bound is the one derived by Caragiannis at al. [4] for the restricted case of load balancing on identical servers, which poses $\textsc{Sum}(S^n) \geq 4\textsc{Opt}_{sum} - \epsilon$, for any $\epsilon > 0$. We close this gap by proving that, for any $\epsilon > 0$, there always exist a linear congestion game and a one-round walk $(S^0, S^1, \ldots, S^n)$, with $S^0 = \emptyset$, such that $\textsc{Sum}(S^n) \geq (2 + \sqrt{5} - \epsilon)\textsc{Opt}_{sum}$.

Given three positive integers n, k and o, with $n \geq 2k + o - 1$ and $k \geq 2o$, we define the game $\mathcal{C}_{n,k,o}$ in which there are n players, $m = n + 1$ resources and each player $i \in [n]$ possesses exactly two strategies s_i and s'_i defined according to the following scheme.

- $s_i = \{e_i\}$ and $s'_i = \{e_{i+1}\} \cup \bigcup_{j=k+1}^{k+i} \{e_j\}$, for any $i \in [k-1]$;
- $s_k = \{e_k\}$ and $s'_k = \bigcup_{j=k+1}^{2k} \{e_j\}$;
- $s_i = \bigcup_{j=k+1}^{i} \{e_j\}$ and $s'_i = \bigcup_{j=i+1}^{k+i} \{e_j\}$, for any $k + 1 \leq i \leq k + o$;
- $s_i = \bigcup_{j=i-o+1}^{i} \{e_j\}$ and $s'_i = \bigcup_{j=i+1}^{\min\{k+i,m\}} \{e_j\}$, for any $k + o + 1 \leq i \leq n$.

A small example in which $n = 22$, $k = 8$ and $o = 3$ is shown in Figure 1.

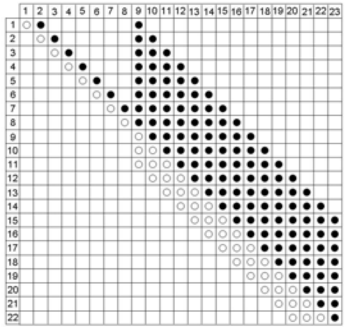


Fig. 1. The set of strategies available to each player in the game $\mathcal{C}_{22,8,3}$. Rows are associated with players, while columns with resources. White and black circles represent the first and the second strategy, respectively.

For any $j \in [m]$ we associate the linear latency function $f_j(x) = a_j \cdot x$ with resource e_j, where each a_j is obtained as a solution of the following system of linear equations.

$$A = \begin{cases} eq_1 \\ eq_2 \\ \dots \\ eq_n \end{cases}$$

where each eq_i is defined as follows:

- $a_1 - a_2 - a_{k+1} = 0$,
- $2a_i - a_{i+1} - \sum_{j=k+1}^{k+i} ((k+i-j+1)a_j) = 0 \;\; \forall\, i = 2, \dots, k-1$,
- $2a_k - \sum_{j=k+1}^{2k} ((2k-j+1)a_j) = 0$,
- $(k+1) \sum_{j=k+1}^{i} a_j - \sum_{j=i+1}^{m} ((k+i-j+1)a_j) = 0 \;\; \forall\, i \in \{k+1, \dots, k+o\}$,
- $(k+1) \sum_{j=i-o+1}^{i} a_j - \sum_{j=i+1}^{\min\{k+i,m\}} ((k+i-j+1)a_j) = 0 \;\; \forall\, i \in \{k+o+1, \dots, n\}$.

Note that the definition of each equality is such that, for any $i \in [n]$, both strategies are equivalent for player i, provided all players $j < i$ have chosen s'_j and all players $j > i$ have not entered the game yet.

Let B be the $n \times m$ coefficient matrix defining system A. The matrix B generated by the game $\mathcal{C}_{22,8,3}$ is shown in Figure 2.

Let $a = (a_1, \dots, a_m)^T$. In order for our instance to be well defined, we need to prove that there exists at least a strictly positive solution to the homogeneous system $Ba = 0$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	-1							-1														
2		2	-1						-2	-1													
3			2	-1					-3	-2	-1												
4				2	-1				-4	-3	-2	-1											
5					2	-1			-5	-4	-3	-2	-1										
6						2	-1		-6	-5	-4	-3	-2	-1									
7							2	-1	-7	-6	-5	-4	-3	-2	-1								
8								2	-8	-7	-6	-5	-4	-3	-2	-1							
9									9	-8	-7	-6	-5	-4	-3	-2	-1						
10									9	9	-8	-7	-6	-5	-4	-3	-2	-1					
11									9	9	9	-8	-7	-6	-5	-4	-3	-2	-1				
12										9	9	9	-8	-7	-6	-5	-4	-3	-2	-1			
13											9	9	9	-8	-7	-6	-5	-4	-3	-2	-1		
14												9	9	9	-8	-7	-6	-5	-4	-3	-2	-1	
15													9	9	9	-8	-7	-6	-5	-4	-3	-2	-1
16														9	9	9	-8	-7	-6	-5	-4	-3	-2
17															9	9	9	-8	-7	-6	-5	-4	-3
18																9	9	9	-8	-7	-6	-5	-4
19																	9	9	9	-8	-7	-6	-5
20																		9	9	9	-8	-7	-6
21																			9	9	9	-8	-7
22																				9	9	9	-8

Fig. 2. The coefficient matrix B generated by the game $\mathcal{C}_{22,8,3}$

Lemma 1. *The system of linear equations $Ba = 0$ admits a strictly positive solution.*

We claim that the strategy profile in which all players choose the second of their strategies is a possible outcome for a one-round walk starting from the empty strategy profile.

Lemma 2. *For any game $\mathcal{C}_{n,k,o}$, there exists a one-round walk $(S^0, S^1, \ldots, S^n)$ such that $S^0 = \emptyset$ and $S^n = (s'_1, \ldots, s'_n)$.*

Proof. The claim is a direct consequence of the definition of system A. □

For our purposes, we do not have to explicitly solve system A, but only need to prove some properties characterizing its set of solutions. We do this in the next two lemmas.

Lemma 3. *In any solution of system A it holds $a_1 \leq 4 \sum_{j=k+1}^{2k} a_j$.*

Lemma 4. *In any solution of system A it holds*

$$(k+1) \sum_{i=m-o+1}^{m} ((i - m + o)a_i) \leq \frac{k^3}{n - 2k - o + 1} \sum_{i=k+1}^{m-o} a_i.$$

We can now prove our main result.

Theorem 1. *For any $\epsilon > 0$, there exist a linear congestion game $\mathcal{C}_{n,k,o}$ and a one-round walk $(S^0, S^1, \ldots, S^n)$, with $S^0 = \emptyset$, such that* $\textsc{Sum}(S^n) \geq (2 + \sqrt{5} - \epsilon)$ $\textsc{Opt}_{sum}$.

Proof. For a fixed integer $n \gg 0$, set $k = \lfloor \sqrt[4]{n} \rfloor$ and $o = \lfloor \frac{3-\sqrt{5}}{2} k \rfloor$. Note that, for a sufficiently big n, these values are consistent with the definition of $\mathcal{C}_{n,k,o}$ since $n \geq 2k + o - 1$ and $k \geq 2o$.

Consider the sum of all the equations defining system A together with the dummy one $a_1 = a_1$. We obtain the equation

$$\sum_{i=1}^{k} 2a_i + (k+1)o \sum_{i=k+1}^{m} a_i - (k+1) \sum_{i=m-o+1}^{m} ((i-m+o)a_i) = \sum_{i=1}^{k} a_i + \frac{k(k+1)}{2} \sum_{i=k+1}^{m} a_i$$

which yields

$$\sum_{i=1}^{k} a_i = (k+1)\left(\frac{k}{2} - o\right) \sum_{i=k+1}^{m} a_i + (k+1) \sum_{i=m-o+1}^{m} ((i-m+o)a_i). \quad (1)$$

Let $S^* = (s_1 \ldots, s_n)$ be the strategy profile in which all players choose the first of their strategies. Because of Lemma 2, we have that there exists a one-round walk $(S^0, S^1, \ldots, S^n)$ such that $S^0 = \emptyset$ and $S^n = (s'_1, \ldots, s'_n)$. By comparing the social costs of S^n and S^*, we obtain

$$\frac{\textsc{Sum}(S^n)}{\textsc{Opt}_{sum}} \geq \frac{\textsc{Sum}(S^n)}{\textsc{Sum}(S^*)} \geq \frac{\sum_{i=2}^{k} a_i + k^2 \sum_{i=k+1}^{m} a_i}{\sum_{i=1}^{k} a_i + o^2 \sum_{i=k+1}^{m} a_i},$$

where we have exploited the fact that $\textsc{Opt}_{sum} \leq \textsc{Sum}(S^*) \leq \sum_{i=1}^{k} a_i + o^2 \sum_{i=k+1}^{m} a_i$.

By using Equality 1, we get

$$\frac{\textsc{Sum}(S^n)}{\textsc{Opt}_{sum}} \geq \frac{\sum_{i=2}^{k} a_i + k^2 \sum_{i=k+1}^{m} a_i}{\sum_{i=1}^{k} a_i + o^2 \sum_{i=k+1}^{m} a_i} =$$

$$\frac{\left((k+1)\left(\frac{k}{2}-o\right)+k^2\right) \sum_{i=k+1}^{m} a_i + (k+1) \sum_{i=m-o+1}^{m} ((i-m+o)a_i) - a_1}{\left((k+1)\left(\frac{k}{2}-o\right)+o^2\right) \sum_{i=k+1}^{m} a_i + (k+1) \sum_{i=m-o+1}^{m} ((i-m+o)a_i)} \geq$$

$$\frac{\left((k+1)\left(\frac{k}{2}-o\right)+k^2+\frac{k^3}{n-2k-o+1}-4\right) \sum_{i=k+1}^{m} a_i}{\left((k+1)\left(\frac{k}{2}-o\right)+o^2+\frac{k^3}{n-2k-o+1}\right) \sum_{i=k+1}^{m} a_i},$$

where, in the last inequality, we have used Lemmas 3 and 4 together with the fact that for any four positive numbers α, β, γ and δ such that $\alpha \geq \beta$ and $\gamma \geq \delta$, it holds $\frac{\alpha+\delta}{\beta+\delta} \geq \frac{\alpha+\gamma}{\beta+\gamma}$.

For n going to infinity, by considering only the dominant terms, we obtain $\lim_{k\to\infty} \frac{\textsc{Sum}(S^n)}{\textsc{Opt}_{sum}} \geq \lim_{k\to\infty} \frac{k\left(\frac{k}{2}-o\right)+k^2}{k\left(\frac{k}{2}-o\right)+o^2} = \lim_{k\to\infty} \frac{\frac{\sqrt{5}-2}{2}k^2+k^2}{\frac{\sqrt{5}-2}{2}k^2+\frac{7-3\sqrt{5}}{2}k^2} = \lim_{k\to\infty} \frac{\frac{\sqrt{5}}{2}k^2}{\frac{5-2\sqrt{5}}{2}k^2} = \frac{\sqrt{5}}{5-2\sqrt{5}} = 2+\sqrt{5}$, which implies the claim. □

Some numerical results, obtained on particular games, are shown in Figure 3. It is possible to appreciate there that the value of k may be chosen much higher than the bound $\lfloor \sqrt[4]{n} \rfloor$ fixed in the proof of Theorem 1. This is due to the fact that, for the sake of simplicity, the bound proved in Lemma 4 is really far from being tight.

n	k	o	$\frac{SUM(S^n)}{SUM(S^*)}$
70	8	3	4.001152
100	8	3	4.012482
500	80	30	4.185590
700	80	30	4.208719
1000	100	38	4.216734
1500	100	38	4.220854
2000	200	76	4.224342
3000	300	114	4.226854

Fig. 3. Lower bounds on the approximation ratio of the solution achieved after a one-round walk starting from the empty strategy profile in the games $\mathcal{C}_{n,k,o}$ for some particular values of n, k and o

4 Social Function Max

In this section we show that the approximation ratio of the solutions achieved after a one-round walk is $\Theta(\sqrt[4]{n^3})$ when starting from the empty strategy profile and $\Theta(n\sqrt{n})$ when starting from an arbitrary one.

Throughout the section, for simplicity and without loss of generality, we will always assume identical latency functions, i.e., $a_e = 1$ and $b_e = 0$ for every $e \in E$. In fact, given a congestion game $\mathcal{C}$ having latency functions $f_e(x) = a_e \cdot x + b_e$ with integer coefficient $a_e, b_e \in \mathbb{R}_{\geq 0}$, it is possible to obtain an equivalent congestion game $\mathcal{C}'$, having the same set of players and identical latency functions as claimed in [7]. Under such assumption, $c_i(S) = \sum_{e\in s_i} n_e(S)$ and $\textsc{Sum} = \sum_{e\in E} n_e^2(S)$.

For every congestion game, given a one-round walk $\left(S^0, S^1, \ldots, S^n\right)$, if $\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum}$, then a trivial upper bound for $\textsc{Sum}(S^n)$ is the following

$$\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum} \leq \mathrm{H}n \cdot \textsc{Opt}_{max}.$$

The following lemma provides a non-trivial upper bound for $\textsc{Sum}(S^n)$ when $S^0 = \emptyset$.

Lemma 5. *For every congestion game, given a one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$ *where* $S^0 = \emptyset$*, if* $\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum}$*, then* $\textsc{Sum}(S^n) \leq \mathrm{T} \cdot \textsc{Opt}_{max}$*, where* $\mathrm{T} = 2n(\sqrt{\mathrm{H}} + 1)$.

By exploiting the above lemma, we establish an upper bound on the approximation ratio of the solutions achieved after a one-round walk starting from the empty strategy profile.

Lemma 6. *For every congestion game, given a one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$ *where* $S^0 = \emptyset$*, if* $\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum}$*, then for any player* k*,* $c_k(S^k) \leq (\sqrt{\mathrm{T}} + 1)\textsc{Opt}_{max}$*, where* $\mathrm{T} = 2n(\sqrt{\mathrm{H}} + 1)$.

Theorem 2. *For every congestion game, given a one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$ *where* $S^0 = \emptyset$*, if* $\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum}$*, then* $\textsc{Max}(S^n) \leq \left(\sqrt{(\sqrt{\mathrm{T}} + 1)\mathrm{T}}\right) \textsc{Opt}_{max}$*, where* $\mathrm{T} = 2n(\sqrt{\mathrm{H}} + 1)$.

Proof. Let k be the player maximizing the cost in S^n, i.e., $c_k(S^n) = \sum_{e \in s'_k} n_e(S^n) = \textsc{Max}(S^n)$. It is easy to see, by applying Cauchy-Schwarz inequality, that

$$\textsc{Max}(S^n) = \sum_{e \in s'_k} n_e(S^n) \leq \sqrt{| s'_k | \sum_{e \in s'_k} n_e^2(S^n)} \leq \sqrt{| s'_k | \sum_{e \in E} n_e^2(S^n)}. \tag{2}$$

Since the number of resources in the strategy of player k cannot be grater than the player's cost, by exploiting Lemma 6 it follows that

$$| s'_k | \leq \left(\sqrt{2n(\sqrt{\mathrm{H}} + 1)} + 1\right) \textsc{Opt}_{max}. \tag{3}$$

From Lemma 5 we know that

$$\sum_{e \in E} n_e^2(S^n) \leq 2n \left(\sqrt{\mathrm{H}} + 1\right) \textsc{Opt}_{max}. \tag{4}$$

By using 3 and 4 in 2, we get

$$\textsc{Max}(S^n) \leq \left(\sqrt{\left(\sqrt{2n\left(\sqrt{\mathrm{H}} + 1\right)} + 1\right) 2n \left(\sqrt{\mathrm{H}} + 1\right)}\right) \textsc{Opt}_{max}. \qquad \square$$

Since $\textsc{Sum}(S^n) \leq (2 + \sqrt{5}) \cdot \textsc{Opt}_{sum}$ [8], by applying Theorem 2, we can claim the following result.

Corollary 1. *For every congestion game* $\mathcal{C}$ *and one-round walk* $(S^0, S^1, \ldots, S^n)$ *such that* $S^0 = \emptyset$*, it holds* $\textsc{Max}(S^n) = O(\sqrt[4]{n^3})\textsc{Opt}_{max}$.

As an asymptotically matching lower bound we can prove the following theorem.

Theorem 3. *There exist a congestion game and a one-round walk* $(S^0, S^1, \ldots, S^n)$*, with* $S^0 = \emptyset$*, such that* $\textsc{Max}(S^n) = \Omega(\sqrt[4]{n^3})\textsc{Opt}_{max}$.

As a consequence of these last two results, we can conclude that the approximation ratio of the solutions achieved after a one-round walk starting from the empty strategy profile is $\Theta(\sqrt[4]{n^3})$. It is worth noting that such a bound is only $\sqrt[4]{n}$ times away from the price of anarchy of pure Nash equilibria which is equal to $\Theta(\sqrt{n})$, as proved in [7].

For the case in which the one-round walk may start from an arbitrary strategy profile we can prove the following results establishing an approximation ratio of $\Theta(n\sqrt{n})$.

Theorem 4. *For every congestion game and one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$*, if* $\textsc{Sum}(S^n) \leq \mathrm{H} \cdot \textsc{Opt}_{sum}$*, then* $\textsc{Max}(S^n) \leq n\sqrt{\mathrm{H}} \cdot \textsc{Opt}_{max}$.

Corollary 2. *For every congestion game and one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$*, it holds* $\textsc{Max}(S^n) = O\left(n\sqrt{n}\right)\textsc{Opt}_{max}$.

Theorem 5. *There exist a congestion game and a one-round walk* $\left(S^0, S^1, \ldots, S^n\right)$ *such that* $\textsc{Max}(S^n) = \Omega\left(n\sqrt{n}\right)\textsc{Opt}_{max}$.

5 Open Problems

We have considered the problem of evaluating the approximation ratio of the solutions achieved after a one-round walk in linear congestion games. For the social function Sum, we have given an exact bound for one-round walks starting from the empty strategy profile. For the social function Max, we have given asymptotically tight bounds for one-round walks starting from either the empty or an arbitrary strategy profile. But, still, several questions remain open even when restricting to one-round walks starting from the empty strategy profile. What about the case of singleton linear congestion games? For the social function Sum, [4] shows an upper bound of 4.055 and a lower bound of 4 in the case of identical resources. For heterogeneous ones, nothing is known except for the 4 lower bound coming from the identical case and the 4.24 upper bound coming from the general case of linear congestion games. For the social function Max, no results are known so far. We can prove a lower bound of $1 + \lfloor \log n \rfloor$ for the case of identical resources. Furthermore, what about extensions to weighted players? To the best of our knowledge, the only known result is the upper bound of $(1 + \sqrt{3})^2 \approx 7.46$ given in [8] for general weighted linear congestion games under the social function Sum.

References

1. Ackermann, H., Röglin, H., Vöcking, B.: On the impact of combinatorial structure on congestion games. In: FOCS, pp. 613–622 (2006)
2. Awerbuch, B., Azar, Y., Epstein, A.: The price of routing unsplittable flow. In: STOC, pp. 57–66. ACM, New York (2005)

3. Awerbuch, B., Azar, Y., Epstein, A., Mirrokni, V.S., Skopalik, A.: Fast convergence to nearly optimal solutions in potential games. In: ACM Conference on Electronic Commerce, pp. 264–273. ACM, New York (2008)
4. Caragiannis, I., Flammini, M., Kaklamanis, C., Kanellopoulos, P., Moscardelli, L.: Tight bounds for selfish and greedy load balancing. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 311–322. Springer, Heidelberg (2006)
5. Chien, S., Sinclair, A.: Convergence to approximate nash equilibria in congestion games. In: SODA, pp. 169–178. SIAM, Philadelphia (2007)
6. Christodoulou, G., Koutsoupias, E.: On the price of anarchy and stability of correlated equilibria of linear congestion games. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 59–70. Springer, Heidelberg (2005)
7. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: STOC, pp. 67–73. ACM, New York (2005)
8. Christodoulou, G., Mirrokni, V.S., Sidiropoulos, A.: Convergence and approximation in potential games. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 349–360. Springer, Heidelberg (2006)
9. Yannakakis, M., Johnson, D.S., Papadimitriou, C.H.: How easy is local search? Journal of Computer and System Sciences 37, 79–100 (1988)
10. Fabrikant, A., Papadimitriou, C.H., Talwar, K.: The complexity of pure nash equilibria. In: STOC, pp. 604–612. ACM, New York (2004)
11. Fanelli, A., Flammini, M., Moscardelli, L.: The speed of convergence in congestion games under best-response dynamics. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 796–807. Springer, Heidelberg (2008)
12. Fotakis, D.: Congestion games with linearly independent paths: Convergence time and price of anarchy. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 33–45. Springer, Heidelberg (2008)
13. Fotakis, D., Kontogiannis, S.C., Spirakis, P.G.: Symmetry in network congestion games: Pure equilibria and anarchy cost. In: Erlebach, T., Persinao, G. (eds.) WAOA 2005. LNCS, vol. 3879, pp. 161–175. Springer, Heidelberg (2006)
14. Harks, T., Klim, M., Mohring, R.H.: Characterizing the existence of potential functions in weighted congestion games. Preprint 011-2009 (2009)
15. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. Telecommunication Systems 17(4), 385–409 (2001)
16. Mirrokni, V.S., Vetta, A.: Convergence issues in competitive games. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 183–194. Springer, Heidelberg (2004)
17. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14, 124–143 (1996)
18. Panagopoulou, P.N., Spirakis, P.G.: Algorithms for pure nash equilibria in weighted congestion games. Journal of Experimental Algorithmics 11(2.7) (2006)
19. Rosenthal, R.W.: A class of games possessing pure-strategy nash equilibria. International Journal of Game Theory 2, 65–67 (1973)
20. Skopalik, A., Vöcking, B.: Inapproximability of pure nash equilibria. In: STOC, pp. 355–364. ACM, New York (2008)
21. Vetta, A.: Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In: FOCS, pp. 416–425 (2002)

Nash Equilibria and the Price of Anarchy for Flows over Time*

Ronald Koch and Martin Skutella

TU Berlin, Inst. f. Mathematik, MA 5-2, Str. des 17. Juni 136, 10623 Berlin, Germany
{koch,skutella}@math.tu-berlin.de

Abstract. We study Nash equilibria in the context of flows over time. Many results on *static* routing games have been obtained over the last ten years. In flows over time (also called *dynamic* flows), flow travels through a network over time and, as a consequence, flow values on edges are time-dependent. This more realistic setting has not been tackled from the viewpoint of algorithmic game theory yet; but there is a rich literature on game theoretic aspects of flows over time in the traffic community.

We present a novel characterization of Nash equilibria for flows over time. It turns out that Nash flows over time can be seen as a concatenation of special static flows. The underlying flow over time model is the so-called *deterministic queuing model* that is very popular in road traffic simulation and related fields. Based upon this, we prove the first known results on the price of anarchy for flows over time.

1 Introduction

In a groundbreaking paper, Roughgarden and Tardos [36] (see also Roughgarden's book [35]) analyze the price of anarchy for selfish routing games in networks. Such routing games are based upon a classical static flow problem with convex latency functions on the edges of the network. In a Nash equilibrium, flow particles (infinitesimal flow units) selfishly choose an origin-destination path of minimum latency.

One main drawback of this class of routing games is its restriction to *static* flows. Flow variation over time is, however, an important feature in network flow problems arising in various applications. As examples we mention road or air traffic control, production systems, communication networks (e.g., the Internet), and financial flows; see, e.g., [5,31]. In contrast to static flow models, flow values on edges may change with time in these applications. Moreover, flow does not progress instantaneously but travels at a certain pace through the network which is determined by transit times on the edges. Both temporal features are captured by *flows over time* (sometimes also called *dynamic* flows) which were introduced by Ford and Fulkerson [15,16].

Another crucial phenomenon in many of those applications mentioned above is the variation of time taken to traverse an edge with the current (and maybe

* This work is supported by DFG Research Center Matheon "Mathematics for key technologies" in Berlin.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 323–334, 2009.

also past) flow situation on this edge. The latter aspect induces highly complex dependencies and leads to non-trivial mathematical flow models. For a more detailed account and further references we refer to [5,11,19,25,31,32]. In particular, all of these *flow over time* models have so far resisted a rigorous algorithmic analysis of Nash equilibria and the price of anarchy.

We identify a suitable flow over time model that is based on the following simplifying assumptions. Every edge of a given network has a fixed free flow transit time and a capacity. The capacity of an edge bounds the rate (flow per time unit) at which flow traverses this edge. The free flow transit time denotes the time that a flow particle needs to travel from the tail to the head of the edge. If, at some point in time, more flow wants to traverse an edge than its capacity allows, the flow particles queue up at the end of the edge and wait in line before they actually enter the head node. When a new flow particle wants to traverse an edge, the time needed to arrive at the head thus consists of the fixed free flow transit time plus the waiting time. In the traffic literature, this flow over time model is known as "deterministic queuing model".

Related Literature. As already mentioned above, flows over time with fixed transit times were introduced by Ford and Fulkerson [15,16]. For more details and further references on these classical flows over time we refer, for example, to [14,37].

So far, Nash equilibria for flows over time were mostly studied within the traffic community. Vickrey [42] and Yagar [45] are the first to introduce this topic. Up to the middle of the 1980's, nearly all contributions consider Nash equilibria on given small instances; see, e.g., [42,21,13,24]. Since then, the number of publications in this area has increased rapidly and Nash equilibria where modeled mathematically. Two main models are distinguished: The route-choice-model where a player only chooses an s-t-path for the controlled flow particle and the simultaneous departure-time-route-choice-model where in addition the departure time is also chosen. For a survey see, e.g., [30]. The considered models can be grouped into four categories: mathematical programming (e.g., [23,20]), optimal control (e.g., [33,18]), variational inequalities (e.g., [17,12,34,39,40]), and simulation-based approaches (e.g., [45,26,7,41,6]). Up to now, variational inequalities are the most common formulation for analyzing Nash equilibria in the context of flows over time.

Many models mentioned above use a path-based formulation of flows over time. Therefore they are computationally often intractable. Edge-based formulations are, for example, considered in [2,12,34]. Realistic assumptions on the underlying flow model with respect to traffic are described by Carey [9,10].

In this paper the deterministic queuing model is considered. This model was introduced by Vickrey [42] and later by Hendrickson and Kocur [21]. Smith [38] shows the existence of an equilibrium for this model in a special case. Akamatsu [1,2] presents an edge-based formulation of the deterministic queuing model on restricted single-source-instances. Akamatsu and Heydecker [3] study Braess's paradox for single-source-instances. Braess's paradox [8] states (for static flows) that increasing the capacity of one edge can increase the total cost of all users in a Nash flow. It is well known that this paradox is extendable to the dynamic case. Mounce [27,28] considers the case where the edge capacities can vary over time and states some existence results. Again, it should be mentioned that these results are based on strong assumptions.

Recently Anshelevich and Ukkusuri [4] analyze a discrete model for Nash equilibria in the context of flows over time. They consider how a single splittable flow unit present at source s at time 0 would traverse a network assuming every flow particle is controlled by a different player. The underlying flow model allowed to send a positive amount of flow over an edge at each integral points in time. Moreover the transit times are assumed to be constant.

Our Contribution. In this paper, we characterize and analyze Nash equilibria for flows over time. Although algorithmic game theory is a flourishing area of research (see, e.g., the recent book [29]), network flows over time have not been studied from this perspective in the algorithms community so far. The main purpose of this paper is to make first steps in this relevant direction, present interesting and novel results, and stimulate further interesting research. We consider the deterministic queuing model in networks with a single source and a single sink. A player controls one flow particle and chooses an s-t-path (route-choice-model) but no departure time which is given a priori.

A precise description of a routing game over time and the underlying flow over time model is given in Section 2. The resulting model of Nash equilibria along with several equivalent characterizations is discussed in Section 3. The main technical contribution of this paper is presented in Section 4. Here we show that a Nash equilibrium can be characterized via a sequence of static flows with special properties. The resulting static flow problems are of interest in their own right. The final Section 5 is devoted to results on the price of anarchy. For the important class of shortest paths networks we prove that every Nash equilibrium is a system optimum. Moreover, a Nash flow over time can be computed in polynomial time by a sequence of sparsest cut computations. Surprisingly, the price of anarchy is, in general, unbounded for arbitrary networks.

Due to space limitations, we omit all proofs in this extended abstract and refer to the full version of the paper [22].

2 A Model for Routing Games over Time

In this section we present a model for Nash equilibria in the context of flows over time. First, in Section 2.1 we define a routing game over time showing the game theoretic aspect of the model. Then in Section 2.2 we introduce an appropriate flow over time model which is known as the deterministic queuing model mentioned above.

Throughout the paper we often use the term *flow particle* in order to refer to an infinitesimal flow unit which corresponds to one player and travels along a single path through the network. The terms *flow rate* and *supply rate* both refer to an amount of flow per time unit.

2.1 From Static Routing Games to Routing Games over Time

Consider a network consisting of a directed graph $G := (V, E)$ with node set V and edge set E. Further, there is a source $s \in V$ and a sink $t \in V$. Each flow particle is a player and the strategy set of each player is the set $\mathcal{P}$ of all s-t-paths.

In a static routing game, the players' decisions yield a static s-t-flow μ of value d where d is the given supply at the source s. Moreover, there is a continuous cost (or payoff) function ℓ_P for each path $P \in \mathcal{P}$ such that $\ell_P(\mu)$ is the cost that a player choosing path P has to pay. The static flow $\mu = (\mu_P)_{P\in\mathcal{P}}$ is a Nash flow if, for all $P \in \mathcal{P}$ with $\mu_P > 0$, it holds that $\ell_P(\mu) = \min_{P'\in\mathcal{P}} \ell_{P'}(\mu)$.

The situation is considerably more complicated when we turn to routing games over time. Here we assume that supply, i.e., players, occur at the source node s over time at a fixed rate d. We can thus identify each player with the point in time θ at which its corresponding flow particle originates at the source. In particular, and in contrast to static routing games, players are not identical. The routing decisions of players yield a flow over time $\mu = (\mu_P)_{P\in\mathcal{P}}$ where μ_P is a function determining the flow rate $\mu_P(\theta)$ at which flow enters path P at time θ and it holds that $\sum_{P\in\mathcal{P}} \mu_P(\theta) = d$, for all θ. Thus, also $\ell_P(\mu)$ is a function which assigns a cost $\ell_P(\mu)(\theta)$ to every point in time θ. That is, the cost experienced by a flow particle that originates at the source at time θ and chooses path P is equal to $\ell_P(\mu)(\theta)$.

In this paper we restrict to payoff functions where $\ell_P(\mu)(\theta)$, $P \in \mathcal{P}$, is the time when a flow originating at s at time θ arrives at t. This time depends upon the particular model of flows over time that we consider which is described in Section 2.2 below.

Like in static routing games, a Nash equilibrium is characterized by a flow over time μ where no player has an incentive to change her chosen path in order to reduce her cost.

Definition 1 (Nash Flow over Time). *Let μ be a flow over time determining the routing decisions of the players in a routing game over time. Then, μ is a Nash equilibrium (Nash flow over time) if, for almost all θ and for all $P \in \mathcal{P}$ with $\mu_P(\theta) > 0$, it holds that $\ell_P(\mu)(\theta) = \min_{P'\in\mathcal{P}} \ell_{P'}(\mu)(\theta)$.*

This definition is an immediate generalization of the definition of static Nash flows under the assumption that the payoff functions are continuous. A closer look at Definition 1 shows us that the continuity of the payoff functions ℓ_P is also essential here. We skip further technical details due to space limitations.

2.2 An Appropriate Flow over Time Model

Although Definition 1 is an immediate generalization of static Nash flows, it is still a highly nontrivial problem to come up with an appropriate flow over time model. Here the main issue are the cost functions ℓ_P, $P \in \mathcal{P}$. For static routing games, these cost functions are not given explicitly, but implicitly via edge latency functions. The cost of a path $P \in \mathcal{P}$ is the sum of the latencies

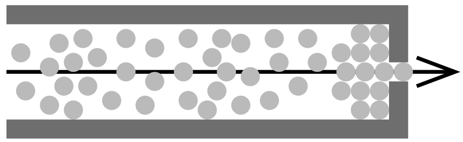


Fig. 1. If more flow particles want to leave an edge than its capacity allows, they form a waiting queue

of its edges. The latency of an edge e is a function of the load μ_e of that edge which can easily be computed as follows: $\mu_e := \sum_{P\in\mathcal{P}:e\in E(P)} \mu_P$.

The situation is considerably more complicated for flows over time. Here, it is usually a highly nontrivial problem to compute the flow rate function μ_e of edge e from given flow rate functions $(\mu_P)_{P\in\mathcal{P}}$. Consider a flow particle that enters a path $P \in \mathcal{P}$ at a certain time θ. Notice that the time at which this particle arrives at an edge $e \in E(P)$ depends on the latencies experienced on the predecessor edges on path P. This fact induces involved dependencies among the flow rate functions $(\mu_e)_{e\in E}$ of the edges. As a consequence, given a flow over time $(\mu_P)_{P\in\mathcal{P}}$, determining the cost (overall latency) of a flow particle entering path P at time θ is, in general, a highly nontrivial task. For more details on this so-called dynamic network loading problem we refer to [43,44]. Nevertheless, for the deterministic queuing model described below, these difficulties can be handled at least for the case of Nash flows over time.

Let (G, u, τ, s, t) be a network consisting of a directed graph $G := (V, E)$, edge capacities $u_e \in \mathbb{R}_+$, $e \in E$, constant *free flow transit times* $\tau_e \in \mathbb{R}_+$, $e \in E$, a source $s \in V$, and a sink $t \in V$. We assume without loss of generality that there are no incoming edges at the source node s and no outgoing edges at the sink node t. The capacity u_e of an edge e bounds the rate at which flow leaves edge e at its head node. The basic concept of the considered flow over time model are waiting queues which built up at the head (exit) of an edge if, at some point in time, more flow particles want to leave an edge than the capacity of the edge allows. The free flow transit time of an edge determines the time for traversing an edge if the waiting queue is empty. Thus, the *(flow-dependent) transit time* on an edge is the sum of the free flow transit time and the current waiting time. We think of the edges as corridors with large entries and small exits, which are wide enough for storing all waiting flow particles (point-queue-model); see Fig. 1.

Every flow particle arriving at an intermediate node v immediately enters the next edge on its path without any delay. In the following we give a more precise mathematical description of the flow over time model. A flow over time is defined by two families of flow rate functions. For an edge e we have an Lebesgue integrable inflow rate f_e^+ meaning that the rate at which flow enters the tail of e at time θ is $f_e^+(\theta) \geq 0$; moreover, the Lebesgue integrable outflow rate f_e^- describes the rate of flow $f_e^-(\theta) \geq 0$ leaving the head of e at time θ. Moreover, we define for an edge e the cumulative in- and outflow at time $\theta \geq 0$ by $F_e^+(\theta) := \int_0^\theta f_e^+(\vartheta)\ d\vartheta$ and $F_e^-(\theta) := \int_0^\theta f_e^-(\vartheta)\ d\vartheta$, respectively. Thus the amount of flow that has entered e before time θ is $F_e^+(\theta)$ and the amount of flow which has traversed e completely before time θ is $F_e^-(\theta)$. Note that F_e^+ and F_e^- are (absolutely) continuous and monotonically increasing, for all $e \in E$.

In order to obtain a feasible flow over time $f := (f^+, f^-)$, the in- and the outflow rates must satisfy several conditions. The capacity of an edge bounds the outflow rate of that edge:

$$f_e^-(\theta) \;\leq\; u_e \qquad\qquad \text{for all } e \in E, \theta \in \mathbb{R}_+. \tag{1}$$

We also have to impose several kinds of flow conservation constraints. Firstly, flow can only traverse an edge if it has previously been assigned to this edge:

$$F_e^+(\theta) - F_e^-(\theta + \tau_e) \;\geq\; 0 \qquad\qquad \text{for all } e \in E, \theta \in \mathbb{R}_+. \tag{2}$$

Secondly, we want flow arriving at an intermediate node $v \in V \setminus \{s, t\}$ to be immediately assigned to an outgoing edge of v:

$$\sum_{e \in \delta^-(v)} f_e^-(\theta) \;=\; \sum_{e \in \delta^+(v)} f_e^+(\theta) \qquad \text{for all } v \in V \setminus \{s, t\}, \theta \in \mathbb{R}_+. \tag{3}$$

In order to ensure that flow which is assigned to an edge must leave this edge again at some point in time, we proceed as follows: Regarding condition (2), the value $F_e^+(\theta)$ is the amount of flow entering edge e before time θ which is equal to the flow arriving at the end of the waiting queue of e until time $\theta + \tau_e$. Moreover, the value $F_e^-(\theta + \tau_e)$ is the amount of flow arriving at the head node of e until time $\theta + \tau_e$. Thus, $F_e^+(\theta) - F_e^-(\theta + \tau_e)$ is the amount of flow in the waiting queue at time $\theta + \tau_e$. We impose the natural condition that, whenever the waiting queue on edge e is nonempty, the flow rate leaving e at its head equals the capacity u_e. Therefore the waiting time spent by a flow particle entering the tail of e at time θ is equal to

$$q_e(\theta) \;:=\; \frac{F_e^+(\theta) - F_e^-(\theta + \tau_e)}{u_e} \qquad\qquad \text{for all } e \in E, \theta \in \mathbb{R}_+. \tag{4}$$

The interpretation of $q_e(\theta)$ as the waiting time for flow particles arriving at time θ on edge e is based on the assumption that the first-in-first-out (FIFO) property holds on edge e. That is, no flow particle overtakes any other flow particle within the waiting queue. Since the free flow transit times are constant, the FIFO property holds for the entire edge.

We state the following proposition which follows directly from (4) and the continuity of F_e^+ and F_e^-.

Proposition 2. *For any edge $e \in E$, the function $\theta \mapsto \theta + q_e(\theta)$ is monotonically increasing and continuous.*

3 Characterizing Nash Flows over Time

The main aspect of Nash equilibria in flow models is the selfish routing of flow particles which are identified with players. As mentioned in Section 2.1, we assume that flow occurs at the source s according to a fixed supply rate $d \in \mathbb{R}_+$.

As soon as a flow particle pops up at the source, it decides by itself how to travel to the sink t. That is, it chooses an s-t-path and immediately enters the first edge on that path.

We consider two classes of flows over time. In the first class, every flow particle travels along "currently shortest paths" only. In the second class, every flow particle tries to overtake as many other flow particles as possible while not be overtaken by others. It turns out that the latter condition leads to a flow where no particle overtakes any other particle. Moreover, we show that the two classes of flows over time coincide and are, in fact, Nash flows over time.

We start by defining *currently shortest s-t-paths* in a given flow over time. To do so, we consider the problem of sending an additional flow particle at time $\theta \geq 0$ from the source s to the sink t as quickly as possible. Let $\ell_v(\theta)$ be the earliest point in time at which this flow particle can arrive at node $v \in V$. Then,

$$\ell_v(\theta) + \tau_e + q_e(\ell_v(\theta)) \;\geq\; \ell_w(\theta) \qquad\qquad \text{for each } e = vw \in E. \tag{5}$$

On the other hand, for each node $w \in V \setminus \{s\}$, there exists at least one incoming edge $e = vw \in \delta^-(w)$ such that equality holds in (5). That is, the flow particle can use edge e in order to arrive at node w as early as possible (at time $\ell_w(\theta)$). Moreover, we have $\ell_s(\theta) = \theta$ for all $\theta \geq 0$. Therefore, we define the *label functions* $\ell_w : \mathbb{R}_+ \to \mathbb{R}_+$ as follows:

$$\ell_w(\theta) \;:=\; \begin{cases} \theta & \text{for } w = s, \\ \min\limits_{e=vw} \ell_v(\theta) + \tau_e + q_e(\ell_v(\theta)) & \text{for } w \in V \setminus \{s\}. \end{cases} \tag{6}$$

The label functions can be computed simultaneously for each time θ by adapting the shortest path algorithm of Bellman and Ford[1]. The following proposition follows from (6) and Proposition 2.

Proposition 3. *For each node $v \in V$, the label function ℓ_v is monotonically increasing and continuous.*

In a Nash equilibrium, flow should always be sent over currently shortest s-t-paths only. We say that edge $e \in E$ *is contained in a shortest path at time* $\theta \geq 0$ if and only if $\ell_w(\theta) = \ell_v(\theta) + \tau_e + q_e(\ell_v(\theta))$. Of course, if an edge $e = vw \in E$ does not lie on a shortest s-t-path at a certain time $\theta \geq 0$, then no flow should be assigned to that edge at time $\ell_v(\theta)$ in a Nash flow.

Definition 4. *We say that* flow is only sent along currently shortest paths *if, for each edge $e = vw \in E$, the following condition holds for almost all times $\theta \geq 0$:*

$$\ell_w(\theta) \;<\; \ell_v(\theta) + \tau_e + q_e(\ell_v(\theta)) \quad\Longrightarrow\quad f_e^+(\ell_v(\theta)) \;=\; 0\ .$$

[1] The update procedure of Bellman-Ford for a certain label $\ell_w(\theta)$ is applied for all times θ simultaneously and, hence, is seen as a operation on functions. If we use Dijkstra instead we have to maintain the set of already finalized nodes separately for each time θ. Thus, we also have to apply the update procedure of Dijkstra separately for each θ.

We emphasize the following aspect of Definition 4: In general, it is not clear that the label functions are *strictly* monotonically increasing. In particular, the label function of the sink t might possibly be constant over a certain time interval $[\theta_1, \theta_2]$ with $\theta_1 < \theta_2$. Thus, a flow particle originating at s at time θ_1 might arrive at t at the earliest possible time without necessarily being as early as possible at all intermediate nodes of its path. Definition 4 enforces, however, that all subpaths of the s-t-path chosen by a flow particle have to be as short as possible.

The condition in Definition 4 is equivalent to the condition that every particle tries to overtake as much other flow as possible while not being overtaken. The latter condition is in fact a *universal FIFO condition*. That is, it is equivalent to the statement that no flow particle can possibly overtake any other flow particle.

In order to model the universal FIFO condition more formally, we consider again an additional flow particle originating at s at time $\theta \geq 0$. Of course, in order to ensure that no flow particle has the possibility to overtake this particle, it is necessary to take a shortest s-t-path. Therefore, for each edge $e = vw \in E$, we define the amount of flow $x_e^+(\theta)$ assigned to e before this particle can reach v and the amount of flow $x_e^-(\theta)$ leaving e before this particle can reach w as follows:

$$x_e^+(\theta) \;:=\; F_e^+(\ell_v(\theta)), \qquad x_e^-(\theta) \;:=\; F_e^-(\ell_w(\theta)) \qquad \text{for all } \theta \geq 0. \tag{7}$$

Thus, the amount of flow $b_s(\theta) := d \cdot \theta$ that has originated at s before our flow particle occurs at s and the amount of flow $-b_t(\theta)$ arriving at t before our flow particle can reach t satisfy

$$b_s(\theta) \;=\; \sum_{e \in \delta^+(s)} x_e^+(\theta) \qquad \text{and} \qquad b_t(\theta) \;=\; -\sum_{e \in \delta^-(t)} x_e^-(\theta)\;. \tag{8}$$

By definition, $b_s(\theta)$ is always nonnegative and $b_t(\theta)$ is always non-positive. If $b_s(\theta) > -b_t(\theta)$, then the considered flow particle overtakes other flow particles. And if $b_s(\theta) < -b_t(\theta)$, then the flow particle is overtaken by other flow particles. This motivates the following definition.

Definition 5. *We say that* no flow overtakes any other flow *if, for each point in time $\theta \geq 0$, it holds that $b_s(\theta) = -b_t(\theta)$.*

Now we are able to prove the equivalence of the universal FIFO condition and the condition that flow only uses currently shortest paths. In addition, a further equivalent statement is given.

Theorem 6. *For a given flow over time, the following statements are equivalent:*

(i) Flow is only sent along currently shortest paths.
(ii) For each edge $e \in E$ and at all times $\theta \geq 0$, it holds that $x_e^+(\theta) = x_e^-(\theta)$.
(iii) No flow overtakes any other flow.
(iv) It is a Nash flow over time.

Note that whenever one of the four statements in Theorem 6 holds, then x^+ and x^- coincide. Further, for all $\theta \geq 0$, setting $x_e(\theta) := x_e^+(\theta)$ for all $e \in E$, yields a static s-t-flow $x(\theta)$ of value $b_s(\theta)$. In the following, for a flow over time satisfying the universal FIFO condition, we refer to $(x_e(\theta))_{e \in E}$ as the *underlying static flow at time θ*. This flow will be studied in more detail in the next section.

4 A Special Class of Static Flows

In this section we study the underlying static flows of a Nash flow over time. It turns out, that these static flows have a special structure that can be used to characterize, compute, and analyze Nash flows over time. Further, the network on which these flows are considered is a special subnetwork of the original network.

Definition 7 (Current Shortest Paths Network). *Consider a flow over time on a network* (G, u, s, t, τ, d)*. For* $\theta \geq 0$*, the* current shortest paths network G_θ *is the subnetwork induced by the edges occurring in a currently shortest path.*

Definition 8 (Thin Flow with Resetting). *Let* (G, u, s, t, d) *be a* static *network and* $E_1 \subseteq E(G)$ *a subset of edges. A* static *flow* x' *with flow value* F *is a* thin flow with resetting on E_1 *if there exist node labels* ℓ' *such that:*

$$\ell'_s = F/d \tag{9}$$

$$\ell'_w \leq \ell'_v \quad \text{for all } e = vw \in E(G) \setminus E_1 \text{ with } x'_e = 0 \tag{10}$$

$$\ell'_w = \max\{\ell'_v, x'_e/u_e\} \quad \text{for all } e = vw \in E(G) \setminus E_1 \text{ with } x'_e > 0 \tag{11}$$

$$\ell'_w = x'_e/u_e \quad \text{for all } e = vw \in E_1 \tag{12}$$

Notice that, if $E_1 = \emptyset$, the label ℓ'_v of node v is the congestion of all flow-carrying s-v-path and a lower bound on the congestion of any s-v-path. Here, the congestion of a path is the maximum congestion of its edges. The name "thin flow *with resetting*" refers to the special edges in E_1 which play the following role. Whenever a path starting at s traverses an edge $e \in E_1$, it "forgets" the congestion of all edges seen so far and "resets" its congestion to x'_e/u_e. It is not difficult to see that, for the special case $E_1 = \emptyset$, a thin flow with resetting can be computed in polynomial time.

Next we show that for a Nash flow over time, the derivatives of the label functions and of the underlying static flow define a thin flow with resetting. The following theorem is only applicable if the derivatives of the label and the underlying static flow functions exist. But both the label functions and the underlying static flow functions are monotonically increasing implying that both families of functions are differentiable almost everywhere.

Theorem 9. *Consider a Nash flow over time on a network* (G, u, s, t, τ, d) *with corresponding label functions* $(\ell_v)_{v \in V}$ *and edge waiting time functions* $(q_e)_{e \in E}$*. For* $\theta' \geq 0$*, let* $x(\theta')$ *be the underlying static flow. Let* $\theta' \geq 0$ *such that* $\frac{dx_e}{d\theta}(\theta')$ *and* $\frac{d\ell_v}{d\theta}(\theta')$ *exist for all* $e \in E$ *and* $v \in V$*. Then, on the current shortest paths network* $G_{\theta'}$*, the derivatives* $(\frac{dx_e}{d\theta}(\theta'))_{e \in E(G_{\theta'})}$ *form a thin flow of value* d *with resetting on the* waiting edges $E_1 := \{e \in E \mid q_e(\theta') > 0\}$*. A corresponding set of node labels fulfilling (9) to (12) is given by the derivatives* $(\frac{d\ell_v}{d\theta}(\theta))_{v \in V(G_\theta)}$*.*

The reverse direction of Theorem 9 also holds. Whenever the derivatives of the underlying static flow functions and the label functions of a flow over time are thin flows with resetting in the current shortest paths network for all times θ, then the flow over time is in fact a Nash flow over time. We skip further details due to space limitations.

5 Nash Flows over Time and the Price of Anarchy

The characterization of Nash flows over time via thin flows with resetting enables us to completely analyze shortest paths networks where every s-t-path has the same total free flow transit time. An important subclass of shortest paths networks are networks where the free flow travel times of all edges are zero. We study the price of anarchy which, in general, is the worst case ratio of the cost of a Nash equilibrium to the cost of a system optimum. In the context of routing games over time, we define the price of anarchy of an instance as the worst case ratio over all points in time θ regarding the following objective:[2] For given θ, maximize the amount of flow arriving at the sink until time θ. In particular, according to this definition, earliest arrival flows that maximize the amount of flow at the sink simultaneously for each point in time are the system optima.

Theorem 10. *For shortest paths networks, each Nash flow over time is an earliest arrival flow and thus a system optimum. Moreover, a Nash flow over time can be computed in polynomial time.*

In contrast to static routing games, there exist instances of the routing game over time where the price of anarchy is unbounded.

Proposition 11. *There exists a family of instances for which the price of anarchy is $\Omega(m)$ where m is the number of edges.*

Acknowledgments. The authors are much indepted to Jose Correa for inspiring discussions on the topic of this paper.

References

1. Akamatsu, T.: A dynamic traffic equilibrium assignment paradox. Transportation Research Part B: Methodological 34(6), 515–531 (2000)
2. Akamatsu, T.: An efficient algorithm for dynamic traffic equilibrium assignment with queues. Transportation Science 35(4), 389–404 (2001)
3. Akamatsu, T., Heydecker, B.: Detecting dynamic traffic assignment capacity paradoxes in saturated networks. Transportation Science 37(2), 123–138 (2003)
4. Anshelevich, E., Ukkusuri, S.: Equilibria in dynamic selfish routing. In: Mavronicolas, M. (ed.) Proceedings of the 2nd International Symposium on Algorithmic Game Theory. LNCS. Springer, Heidelberg (2009)
5. Aronson, J.E.: A survey of dynamic network flows. Annals of Operations Research 20(1–4), 1–66 (1989)
6. Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., Nagel, K.: MATSim-T: Architecture and simulation times. In: Bazzan, A., Klügl, F. (eds.) Multi-Agent Systems for Traffic and Transportation Engineering, ch. III. Information Science Reference (2009)
7. Ben-Akiva, M.J., Bierlaire, M., Koutsopoulos, H.N., Mishalani, R.: DynaMIT: a simulation-based system for traffic prediction and guidance generation. In: Proceedings of the 3rd Triennial Symposium on Transportation Systems (1998)

[2] This objective is well motivated if we think of, e.g., modeling an evacuation situation.

8. Braess, D.: Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung 12 12, 258–268 (1968) (in German)
9. Carey, M.: Link travel times I: Properties derived from traffic-flow models. Networks and Spatial Economics 4(3), 257–268 (2004)
10. Carey, M.: Link travel times II: Properties derived from traffic-flow models. Networks and Spatial Economics 4(4), 379–402 (2004)
11. Chen, H.-K.: Dynamic Travel Choice Models: A Variational Inequality Approach. Springer, Berlin (1999)
12. Chen, H.K., Hsueh, C.F.: A model and an algorithm for the dynamic user-optimal route choice problem. Transportation Research Part B: Methodological 32(3), 219–234 (1998)
13. de Palma, A., Ben-Akiva, M., Lefèvre, C., Litinas, N.: Stochastic equilibrium model of peak period traffic congestion. Transportation Science 17(4), 430–453 (1983)
14. Fleischer, L.K., Tardos, É.: Efficient continuous-time dynamic network flow algorithms. Operations Research Letters 23(3-5), 71–80 (1998)
15. Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. Operations Research 6(3), 419–433 (1958)
16. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
17. Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., Wie, B.W.: A variational inequality formulation of the dynamic network user equilibrium problem. Operations Research 41(1), 179–191 (1993)
18. Friesz, T.L., Luque, J., Tobin, R.L., Wie, B.W.: Dynamic network traffic assignment considered as a continuous time optimal control problem. Operations Research 37(6), 893–901 (1989)
19. Hamacher, H.W., Tjandra, S.A.: Mathematical modelling of evacuation problems: A state of the art. In: Schreckenberg, M., Sharma, S.D. (eds.) Pedestrian and Evacuation Dynamics, pp. 227–266. Springer, Berlin (2002)
20. Han, S., Heydecker, B.G.: Consistent objectives and solution of dynamic user equilibrium models. Transportation Research Part B: Methodological 40(1), 16–34 (2006)
21. Hendrickson, C., Kocur, G.: Schedule delay and departure time decisions in a deterministic model. Transportation Science 15(1), 62–77 (1981)
22. Koch, R., Skutella, M.: Nash equilibria and the price of anarchy for flows over time. Technical Report 014-2009, TU Berlin (2009), `http://www.math.tu-berlin.de/coga/publications/techreports/2009/Report-%014-2009.xhtml`
23. Lin, W.H., Lo, H.K.: Are the objective and solutions of dynamic user-equilibrium models always consistent? Transportation Research Part A: Policy and Practice 34(2), 137–144 (2000)
24. Mahmassani, H., Herman, R.: Dynamic user equilibrium departure time and route choice on idealized traffic arterials. Transportation Science 18(4), 362–384 (1984)
25. Mahmassani, H.S., Peeta, S.: System optimal dynamic assignment for electronic route guidance in a congested traffic network. In: Gartner, N.H., Improta, G. (eds.) Urban Traffic Networks. Dynamic Flow Modelling and Control, pp. 3–37. Springer, Berlin (1995)
26. Mahmassani, H.S., Sbayti, H.A., Zhou, X.: DYNASMART-P Version 1.0 Users Guide. Maryland Transportation Initiative, College Park, Maryland (2004)
27. Mounce, R.: Convergence in a continuous dynamic queueing model for traffic networks. Transportation Research Part B: Methodological 40(9), 779–791 (2006)
28. Mounce, R.: Convergence to equilibrium in dynamic traffic networks when route cost is decay monotone. Transportation Science 41(3), 409–414 (2007)

29. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
30. Peeta, S., Ziliaskopoulos, A.K.: Foundations of dynamic traffic assignment: The past, the present and the future. Networks and Spatial Economics 1(3–4), 233–265 (2001)
31. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.) Network Routing. Handbooks in Operations Research and Management Science, ch. 3, vol. 8, pp. 141–295. North-Holland, Amsterdam (1995)
32. Ran, B., Boyce, D.E.: Modelling Dynamic Transportation Networks. Springer, Berlin (1996)
33. Ran, B., Boyce, D.E., Leblanc, L.J.: A new class of instantaneous dynamic user-optimal traffic assignment models. Operations Research 41(1), 192–202 (1993)
34. Ran, B., Hall, R.W., Boyce, D.E.: A link-based variational inequality model for dynamic departure time/route choice. Transportation Research Part B: Methodological 30(1), 31–46 (1996)
35. Roughgarden, T.: Selfish Routing and the Price of Anarchy. MIT Press, Cambridge (2005)
36. Roughgarden, T., Tardos, É.: How bad is selfish routing? In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, pp. 93–102 (2000)
37. Skutella, M.: An introduction to network flows over time. In: Cook, W., Lovász, L., Vygen, J. (eds.) Research Trends in Combinatorial Optimization, ch. 21, pp. 451–482. Springer, Berlin (2009)
38. Smith, M.J.: The existence of a time-dependent equilibrium distribution of arrivals at a single bottleneck. Transportation Science 18(4), 385–394 (1984)
39. Smith, M.J.: A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. Transportation Research Part B: Methodological 27(1), 49–63 (1993)
40. Szeto, W.Y., Lo, H.K.: A cell-based simultaneous route and departure time choice model with elastic demand. Transportation Research Part B: Methodological 38(7), 593–612 (2004)
41. Taylor, N.B.: The CONTRAM dynamic traffic assignment model. Networks and Spatial Economics 3(3), 297–322 (2003)
42. Vickrey, W.S.: Congestion theory and transport investment. The American Economic Review 59(2), 251–260 (1969)
43. Wu, J.H., Chen, Y., Florian, M.: The continuous dynamic network loading problem: a mathematical formulation and solution method. Transportation Research Part B: Methodological 32(3), 173–187 (1998)
44. Xu, Y.W., Wu, J.H., Florian, M., Marcotte, P., Zhu, D.L.: Advances in the continuous dynamic network loading problem. Transportation Science 33(4), 341–353 (1999)
45. Yagar, S.: Dynamic traffic assignment by individual path minimization and queuing. Transportation Research 5(3), 179–196 (1971)

Bayesian Auctions with Friends and Foes

Po-An Chen and David Kempe

Department of Computer Science
University of Southern California
{poanchen,dkempe}@usc.edu

Abstract. We study auctions whose bidders are embedded in a social or economic network. As a result, even bidders who do not win the auction themselves might derive utility from the auction, namely, when a friend wins. On the other hand, when an enemy or competitor wins, a bidder might derive negative utility. Such spite and altruism will alter the bidding strategies. A simple and natural model for bidders' utilities in these settings posits that the utility of a losing bidder i as a result of bidder j winning is a constant (positive or negative) fraction of bidder j's utility.

We study such auctions under a Bayesian model in which all valuations are distributed independently according to a known distribution, but the actual valuations are private. We describe and analyze Nash Equilibrium bidding strategies in two broad classes: regular friendship networks with arbitrary valuation distributions, and arbitrary friendship networks with identical uniform valuation distributions.

1 Introduction

The traditional view of auctions posits that bidders only care if they win the item(s) and at what price. The utility of bidders not winning the auction is 0, regardless of the actual outcome. If the auction is conducted among perfectly rational strangers, the items are solely for resale, and no future competitive advantage is gained by winning an auction cheaply, this assumption is quite accurate. However, in many realistic scenarios, the bidders are embedded in social and economic networks, which will affect their perception of an auction's outcome.

For instance, if bidders i and j are friends, and bidder j wins the auction at a cheap price, bidder i may derive direct or indirect benefits. These could take the form of shared joy, or tangible financial benefits due to bidder j's generosity. Similarly, if bidders i and j are enemies, then bidder i might actively resent j's winning. Even in a purely economic environment, there can be differences in i's perception of the winner. For instance, if j is a direct competitor of i, then winning the auction cheaply might give bidder j significant future market advantage. Thus, bidder i derives negative utility from j's winning. If j mostly belongs to a different market, then i might be neutral to j's winning. And if i and j might be future collaborators, then i might derive some positive utility from j's winning.

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 335–346, 2009.

These observations motivate the study of auctions in which the utility of losers is not always 0, but rather depends on the *identity of the winner, and the utility the winner derives from the auction*. We can model this setting naturally with a spite/altruism matrix $A = (a_{i,j})$, where each $a_{i,j} \in (-1, 1)$ for $i \neq j$, and $a_{i,i} \in [0, 1]$ for each i. If bidder j wins the auction and obtains subutility $\overline{u}_j$, then bidder i's utility from the auction is $a_{i,j} \cdot \overline{u}_j$. Thus, if $a_{i,j} > 0$, then bidder i is *altruistic* toward bidder j (or a *friend*); if $a_{i,j} < 0$, then bidder i is *spiteful* toward bidder j (or a *foe*)[1]. Notice that we do not assume A to be symmetric.

Auctions with spite among bidders have been studied before [6,7,16,18,21]. However, in all past work, the assumption was that each off-diagonal entry of the matrix A was the same (and negative), i.e., all bidders have the same spite level toward each other. We call this the case of *uniform spite*. While it is interesting as an analysis of the effects of general distrust or future competition between bidders, it does not take into account the effects of social or economic networks on individual behaviors.

In this paper, we study auctions with Bayesian priors in the presence of more general altruism/spite matrices. For two large subclasses of these auctions, we explicitly describe a Nash Equilibrium. These two subclasses are the following:

1. The spite/altruism matrix A is arbitrary, but each bidder's valuation of the item is drawn independently and uniformly from the interval $[0, 1]$, and the auction is first-price.
2. The valuations are drawn independently from $[0, 1]$ according to an arbitrary (but identical) distribution for all bidders, and the social network of bidders is *regular*. This means that each bidder i has non-zero $a_{i,j}$ for the same number d of other bidders j, and all such non-zero entries have the same value $a_{i,j} = a$. In this case, we analyze both first- and second-price auctions.

These characterizations significantly generalize recent results of Morgan et al. [18] and Brandt et al. [6], which characterized a Nash Equilibrium for the case of uniform spite. We also point out here that the equilibrium in the first case is not symmetric: different bidders have different bidding strategies. This is significant in particular from a technical viewpoint, as past analysis has relied heavily on symmetry assumptions in order to be able to derive equilibrium strategies.

Our explicit characterization allows us to derive several interesting corollaries. For the case of arbitrary social networks and uniform valuation distributions, the explicit characterization allows us to study how changes in the social or economic network affect bidding behavior. Perhaps somewhat surprisingly, an increase in spite does not always lead to an increase in bids. Instead, we show that whether it leads to an increase or decrease in bids depends on whether the recipient of spite is currently overbidding or underbidding.

A further corollary concerns auctions with several cliques of friends who are indifferent to other cliques. Our characterization allows us to easily derive explicit equilibrium bidding strategies in this case. Interestingly, the strategy of the

[1] Past work [6,7] used $a_{i,j} > 0$ for spite. However, the convention we adopt here simplifies notation, and is consistent with the notation of [14,9,17].

members of a clique depends only on the size of the clique, the strength of ties in the clique, and the total number of bidders, but not on the strength of the ties in other cliques. In this case in particular, the resulting bidding strategies can be considered an alternative to collusion without explicit information exchange between the members of a clique.

For the case of regular social networks and uniformly random valuations, we show that if $a < 0$ (i.e., bidders only have foes and neutral other bidders), the expected revenue of the second-price auction dominates the first-price auction. Conversely, if $a > 0$, then the expected revenue of the first-price auction dominates the second-price auction. (The case of $a = 0$ corresponds exactly to standard auctions, and the Revenue Equivalence Theorem implies that both auctions provide the same revenue.)

1.1 Related Work

The notion of *spite* and *altruism* as defined here broadly falls into the class of *allocation externalities* in auctions: the utility of a bidder depends not exclusively on her own allocation, but also on the allocations of other bidders. There is a large amount of literature on various types of allocation externalities (see, e.g., [11,12,13,8]). In particular, Jehiel et al. [12] construct revenue-maximizing auctions for the case where each potential buyer has a given constant externality depending on the identity of the winner. Thus, the difference to our model is that in the model of [12], a loser's utility does not depend on the *price* at which the winner obtained the object, only the winner's identity.

Altruism and spite specifically in the context of auctions were studied in several recent papers: Brandt and Weiß [7] studied full-information equilibria between two bidders, both of whom have spite level $a = \frac{1}{2}$. Morgan et al. [18] and Brandt et al. [6] focused on Bayesian Nash Equilibria of first-price and second-price auctions with uniform spite. The results in these two papers are very similar to each other, and differ mostly in the precise model of the utility of the winner, as discussed briefly in Section 2. Vetsikas and Jennings [21] extend this work to auctions for multiple items, still assuming uniform spite among the bidders.

A similar model is also studied in a recent paper by Deng and Qi [10] on auction design for pricing priority rights. Losers in this model also incur a negative utility, albeit one that depends on their own utility for the item, rather than the winner's. The goal in [10] is to design a truthful, egalitarian and budget-balanced auction.

Several recent papers have analyzed the impact of spiteful or altruistic behavior in other game-theoretic settings. In the context of congestion games, [9] shows that when all players are at least β-altruistic (meaning that their utility is a convex combination of their own latency and the derivative of the average latency, with weight β on the average latency), then the PoA is bounded by $1/\beta$. Babaioff et al. [3] and Roth [20] consider the effect of malicious or Byzantine players on the PoA or regret.

In the context of network inoculation [2], Moscibroda et al. [19] study the effect of Byzantine malicious players, while Meier et al. [17] show that friendship with neighbors can sometimes lead to significantly more efficient network inoculations.

The impact of social network structure on games has also recently been studied by Ashlagi et al. [1], under the name *social context games*. They posit that the utility of an agent can be computed from the subutility functions in her neighborhood, according to various competitive or collaborative aggregation functions. The specific games studied in [1] differ from the auctions considered here, and mostly belong to the class of resource selection games.

2 Model and Preliminaries

Each of the n bidders has a valuation v_i drawn independently from *the same* distribution F over $[0, 1]$. While the valuations are private, F is common knowledge among all bidders. We identify the distribution with its cumulative distribution function (cdf), and use $f = F'$ to denote its density function.

We study auctions in which the auctioneer is selling a single item to spiteful and altruistic bidders. Bids are denoted by b_i. The auction mechanism selects as winner a bidder w maximizing b_w (breaking ties arbitrarily, but consistently). We define the *threshold bid* of w to be $\tau_w = \max_{j \neq w} b_j$. In a first-price auction, bidder w pays b_w, while in a second-price auction, she pays τ_w.

In a second-price auction, the *subutility* of the winning bidder is $\overline{u}_w = v_w - \tau_w$. Similarly, for a first-price auction, the subutility of the winning bidder is $\overline{u}_w = v_w - b_w$. In both cases, the subutility of all losing bidders is $\overline{u}_i = 0$.

In an auction with altruism and spite, the *utility* of a bidder is a combination of her own and the other bidders' subutilities. (A similar model was proposed by Ledyard [14].) Specifically, for any bidder i:

$$u_i = \textstyle\sum_j a_{i,j} \cdot \overline{u}_j. \tag{1}$$

Since it is reasonable to assume that each bidder cares more about her own subutility than that of others, we assume that $|a_{i,j}| < a_{i,i}$ for all i, j. Substituting the specific subutilities of first-price and second-price auctions into Equation (1), we obtain the following utilities for bidders i.

$$\text{First-price auction:} \qquad u_i = \begin{cases} a_{i,i} \cdot (v_i - b_i) & \text{for } i = w \\ a_{i,w} \cdot (v_w - b_w) & \text{for } i \neq w \end{cases} \tag{2}$$

$$\text{Second-price auction:} \qquad u_i = \begin{cases} a_{i,i} \cdot (v_i - \tau_i) & \text{for } i = w \\ a_{i,w} \cdot (v_w - \tau_w) & \text{for } i \neq w \end{cases} \tag{3}$$

Remark 1. The definition used by Brandt et al. [6] is the special case of our definition when $a_{i,j} = a < 0$ for all $i \neq j$, i.e., players have uniform spite, and $a_{i,i} = 1 + a$. (However, [6] uses $a > 0$ for spite.) The definition of Morgan et al. [18] is nearly identical, except it corresponds to the case of $a_{i,i} = 1$ for all i.

Bidders are assumed to maximize expected utility, and may submit bids $b_i \neq v_i$. Specifically, we denote the bid function for bidder i by $b_i(\cdot)$, meaning that with valuation v, bidder i will submit a bid of $b_i(v)$. We stress here that while the valuations are private, both the common distribution of valuations and the altruism/spite matrix A are common knowledge. (We briefly discuss the latter point in Section 4.) The externalities in our setting arise solely from the perception that losers have of the winner; there is no correlation between the valuations of different bidders.

According to Equation (1), the larger $|a_{i,j}|$ (and the smaller $a_{i,i}$), the more important the winning or losing of other bidders becomes to i. Notice, however, that we do not recursively consider the utility a bidder derives from another bidder's perceived utility. Such *systems of utility functions* are studied, for instance, by Bergstrom [4], who shows that by solving a system of linear equations, they can be reduced to the case studied here.

While our model allows for friendship between bidders, we assume that the bidders do not collude. Collusion would require the bidders to share their private valuations with each other, which may be an inferior strategy in terms of the individual utilities. Furthermore, it is not clear how the profit should be split between colluding bidders when $0 < a_{i,j} < 1$.

3 Calculating and Analyzing Equilibria

We first derive general Nash Equilibrium conditions for arbitrary distributions F on $[0, 1]$ and altruism, for both first- and second-price auctions. Since these conditions are too complicated to solve in general, we then focus on two special cases:

1. First-price auctions with arbitrary spite/altruism matrices A, but in which all valuations are drawn *uniformly* from the interval $[0, 1]$. For this case, we present a (non-symmetric) Nash Equilibrium, and show how the bidding strategies change if the entries of A change.
2. Networks in which each bidder has the same number d of acquaintances, and feels the same spite/altruism level a toward all of them. Thus, we have a social network in which each node has outdegree d, and all bidders have uniform spite/altruism. For this case, we analyze both first- and second-price auctions under arbitrary distributions F of valuations. We show that under uniformly random valuations, the revenue of the second-price auction dominates the first-price auction for $a < 0$, while the domination is reversed for $a > 0$.

We denote by b_i^{-1} the inverse function of the bidding function, i.e., $b_i^{-1}(b)$ is the valuation v such that bidder i with valuation v would bid b.[2]

Lemma 1. *Assume that all valuations are drawn independently from the same distribution F over $[0, 1]$.*

1. *Nash Equilibria of first-price auctions satisfy the following system of differential equations:*

$$\sum_{j\neq i} \left(a_{i,i}(v - b_i(v)) + a_{i,j}(b_i(v) - b_j^{-1}(b_i(v)))\right) \cdot \frac{f(b_j^{-1}(b_i(v)))\cdot b_j^{-1\prime}(b_i(v))}{F(b_j^{-1}(b_i(v)))} = a_{i,i}. \quad (4)$$

[2] We are thus implicitly assuming that the bidding functions are strictly increasing.

2. *Nash Equilibria of second-price auctions satisfy the following system of differential equations:*

$$\begin{aligned}
&a_{i,i} \cdot \sum_{j \neq i} \frac{f(b_j^{-1}(b_i(v)) \cdot b_j^{-1\prime}(b_i(v)))}{F(b_j^{-1}(b_i(v)))} \cdot (v - b_i(v)) \\
&+ \sum_{j \neq i} \frac{a_{i,j}}{F(b_j^{-1}(b_i(v)))} \cdot \Big(- F(b_j^{-1}(b_i(v))) \cdot b_i(v) \cdot \sum_{k \neq i,j} \frac{f(b_k^{-1}(b_i(v))) b_k^{-1\prime}(b_i(v))}{F(b_k^{-1}(b_i(v)))} \\
&\qquad + b_i(v) \cdot \sum_{\ell \neq i} f(b_\ell^{-1}(b_i(v))) \cdot b_\ell^{-1\prime}(b_i(v)) \\
&\qquad - b_j^{-1}(b_i(v)) \cdot f(b_j^{-1}(b_i(v))) \cdot b_j^{-1\prime}(b_i(v)) - 1 \Big) \\
= &- \sum_{j \neq i} a_{i,j}
\end{aligned} \tag{5}$$

The proof is rather technical, and deferred to the full version of this paper due to space constraints. Compared to standard analysis of equilibrium strategies in auctions, it requires deriving a system of differential equations, rather than a single differential equation. In general, this system of differential equations (4) or (5) does not admit a direct solution, due to the interplay between inverses of bidding functions. We therefore next focus on special cases where the particular form of bidding functions allows us to simplify the differential equations further.

3.1 First-Price Auctions with Uniform Valuations

Our first special case is that of first-price auctions with uniform valuations on $[0, 1]$, i.e., $F(x) = x$ for $x \in [0, 1]$. In this case, we can calculate a Bayesian Nash Equilibrium explicitly, because there happens to be a Nash Equilibrium where each bidder bids $b_i(v) = \gamma_i v$ for some constant γ_i. Unfortunately, a guess of $b_i(v) = \gamma_i v$, or even $b_i(v) = \gamma_i v + \xi_i$, does not appear to lead to a solution of the corresponding system (5) for second-price auctions, and we are not aware of any explicit characterization of an equilibrium of the second-price auction here.

Theorem 1. *Assume that all valuations are drawn independently and uniformly from* $[0, 1]$. *There is a Bayesian Nash Equilibrium for first-price auctions with an arbitrary friendship/spite matrix A where each bidder i bids $b_i(v_i) = \gamma_i v_i$, with*

$$\gamma_i = \frac{\det(C)}{\det(C) - \det(C_i)}.$$

The matrix C has entries $c_{i,i} = -(n-1)$ and $c_{i,j} = \frac{a_{i,j}}{a_{i,i}}$ for $i \neq j$, and C_i is formed by replacing the i^{th} column of C by all 1's.

Proof. We start with the general system of differential equations derived as Equation (4). We now substitute that $F(x) = x$ and $f(x) = 1$ for all $x \in [0, 1]$, obtaining that

$$\sum_{j \neq i} \Big(a_{i,i}(v - b_i(v)) + a_{i,j}(b_i(v) - b_j^{-1}(b_i(v))) \Big) \cdot \frac{b_j^{-1\prime}(b_i(v))}{b_j^{-1}(b_i(v))} = a_{i,i}.$$

We next guess that $b_i(v) = \gamma_i v$ for each bidder i, i.e., each bidder simply scales her valuation by a constant factor that may depend on A, but not on the valuations. Then, $(b_j^{-1})'(b_i(v)) = 1/\gamma_j$, and $b_j^{-1}(b_i(v)) = \frac{\gamma_i}{\gamma_j} \cdot v$, so we obtain

$$\sum_{j \neq i} \Big(a_{i,i}(1 - \gamma_i)v + a_{i,j}(\gamma_i v - \frac{\gamma_i}{\gamma_j} \cdot v) \Big) \cdot \frac{1}{\gamma_i v} = a_{i,i}.$$

Canceling all v terms and the γ_i, and pulling constant terms out of the sum, this simplifies to

$$(n-1)a_{i,i}(\frac{1}{\gamma_i}-1)+\sum_{j\neq i} a_{i,j}(1-\frac{1}{\gamma_j})=a_{i,i}.$$

Writing $\beta_i = 1-\frac{1}{\gamma_i}$, this system becomes $-(n-1)\beta_i + \sum_{j\neq i}\frac{a_{i,j}}{a_{i,i}}\beta_j = 1$ for all i. Thus, the vector $\boldsymbol{\beta}$ of all β_i entries solves $C\cdot\boldsymbol{\beta}=\mathbf{1}$, where $\mathbf{1}$ is the n-dimensional all-ones vector. The theorem now follows from Cramer's rule, which gives that $\beta_i = \frac{\det(C_i)}{\det(C)}$. Because $|a_{i,j}| < a_{i,i}$ for all i,j, all off-diagonal entries of C are strictly less than 1, so C is diagonally dominant. Gershgorin's Disc Theorem thus guarantees that $\det(C)\neq 0$, and the system always has a solution. ■

Competing Cliques. One natural special case which can be solved easily using our general result in Theorem 1 is that of disjoint cliques of friends in an auction. The bidders form g disjoint groups $S_1,\ldots,S_g$. Within group S_k, all bidders have altruism $a^{(k)}$ to each other (and $a_{i,i}=1$). Across groups, bidders are indifferent, i.e., a bidder's altruism or spite level towards any other bidder who is not in his group is 0. Then, C is a block matrix, and the system of linear equations can be solved for each block separately. Due to symmetry, within each group S_k, all bidders will use the same bidding strategy, i.e., $\beta_i=\beta_j =: \beta^{(k)}$ whenever $i,j\in S_k$. The linear equality thus simplifies to $-(n-1)\beta^{(k)} + (|S_k|-1)\cdot a^{(k)}\cdot\beta^{(k)} = 1$, with the solution $\beta^{(k)} = \frac{1}{(|S_k|-1)\cdot a^{(k)}-(n-1)}$. Substituting this into the definition of γ_i, we obtain the following corollary:

Corollary 1. *If the bidders form disjoint cliques S_k with mutual altruism $a^{(k)}$, and all valuations are drawn uniformly from* $[0,1]$*, then there exists a Bayesian Nash Equilibrium in which each bidder $i\in S_k$ bids* $\frac{n-1-a^{(k)}(|S_k|-1)}{n-a^{(k)}(|S_k|-1)}\cdot v_i$.

Notice that this corollary reveals several interesting tendencies. First, both β_i and γ_i are always less than 1, and decreasing in $|S_k|$ and $a^{(k)}$. This is not entirely unexpected, as bidders in large or tightly knit cliques feel less of a need to win the auction themselves, since they are more likely to derive utility from a friend's winning. What is perhaps more surprising is that the bidding strategy of a clique S_k does not depend on how large or tightly knit another group $S_{k'}$ is. While this follows readily from our general result, it is not at all apparent a priori, since another tightly knit group might bid lower, allowing group S_k to lower its bids safely as well.

Altruism Changes. We can also use Theorem 1 for an investigation of how bidder i's strategy changes if her spite level $a_{i,j}$ toward another member of the network changes. One would intuitively expect that if bidder i's altruism toward bidder j increases, then bidder i will always bid lower, i.e., decrease γ_i, because she derives more utility from bidder j's winning. (Indeed, for disjoint cliques, this intuition is borne out.) It turns out that this is not always the case. In response to the change of one $a_{i,j}$, the entire network's strategies adapt, and in some cases, this means that bidder i will increase her bid. The following theorem characterizes the change — its proof is quite tedious and technical, and deferred to the full version of this paper due to space constraints.

Theorem 2. *The derivative of β_i with respect to $a_{i,j}$ is*

$$\frac{\partial \beta_i}{\partial a_{i,j}} = -\frac{\det(C_{i,i}) \cdot \det(C_j)}{\det(C)^2} = -\frac{\det(C_{i,i})}{\det(C)} \cdot \beta_j,$$

where $C_{i,i}$ is the $(n-1) \times (n-1)$ matrix obtained by removing the i^{th} row and column from C, and C_j is the matrix formed by replacing the j^{th} column of C by all ones.

In the result of Theorem 2, $\det(C_{i,i})$ and $\det(C)$ always have opposite signs, because all their eigenvalues are negative, and one of them has even, the other one odd, dimension. Thus, if bidder i increases her altruism level to another bidder j who is overbidding ($\beta_j > 0$, thus $\gamma_j > 1$), then bidder i will increase her bid. Conversely, if she increases her altruism level to another bidder j who is currently underbidding, then bidder i will decrease her bid. Thus, the current bidding strategy of bidder j captures enough information to determine the direction of the change in bidder i's bid when her altruism or spite changes.

3.2 Regular Networks

In order to be able to solve the system of differential equations, we assumed in the previous section that the valuations were drawn uniformly from $[0, 1]$. As a first step towards avoiding this assumption, we consider *regular networks*, in which each node has the same out-degree d. Furthermore, we assume that for each pair of bidders (i, j) with a directed edge from i to j, the spite level is the same, $a_{i,j} = a$ for all i, j with an edge. Similarly, all diagonal entries are the same, i.e., $a_{i,i} = \alpha$ for all i.

Under this scenario, both the first-price and second-price auction have a symmetric Bayesian Nash Equilibrium, i.e., a Nash Equilibrium in which all bidding functions are the same, $b_i = b$ for all i.

Theorem 3. *For $\alpha \neq 0$, there exists a Bayesian Nash Equilibrium for first-price auctions in which all bidders bid $b(v) = \mathrm{E}[X \mid X < v]$, where X is a random variable with cdf $F(x)^{n-1-da/\alpha}$.*

Proof. Substituting the symmetric guess into the the system (4) for first-price auctions, we can simplify to

$$\sum_{j \neq i} \Big(a_{i,i}(v - b(v)) + a_{i,j} b(v) - a_{i,j} v \Big) \cdot \frac{f(v)}{F(v) b'(v)} = a_{i,i},$$

and, using the network structure, simplify further to

$$\Big(((n-1)\alpha - da) \cdot (v - b(v)) \Big) \cdot \frac{f(v)}{F(v) b'(v)} = \alpha.$$

Solving for $b(v)$ gives us $b(v) = v - \frac{1}{n-1-da/\alpha} \cdot \frac{F(v) b'(v)}{f(v)}$. This differential equation has solution

$$b(v) = F(v)^{-(n-1-da/\alpha)} \cdot \int_0^v x \cdot (n - 1 - da/\alpha) \cdot F(x)^{n-2-da/\alpha} f(x) dx. \qquad (6)$$

Thus, we have proved the theorem. ∎

Note that the bidding function can be interpreted as the expectation of the highest of $(n-1)-\frac{ad}{\alpha}$ private values below v, in spite of the fact that $(n-1)-\frac{ad}{\alpha}$ may be a fractional number. Notice that this theorem does not characterize *all* equilibria, and indeed, it seems very likely that this auction also possesses asymmetric Nash Equilibria (see also the discussion in [6]).

Substituting the uniform distribution over $[0,1]$ for every bidder's valuation, we obtain the following corollary:

Corollary 2. *There is a Bayesian Nash Equilibrium for first-price auctions with all valuations uniformly distributed in* $[0,1]$ *in which all bidders bid* $b(v) = (1 - \frac{\alpha}{n\cdot\alpha - ad}) \cdot v$.

In particular, when $d = n-1$, Theorem 3 and Corollary 2 recover the results of Brandt et al. [6] who showed that $b(v) = \frac{n-1}{n+a} \cdot v$ for uniform spite levels (with $\alpha = 1 + a$), and those of Morgan et al. [18] (with $\alpha = 1$).

Second-Price Auctions. We next turn our attention to second-price auctions, and prove the following theorem.

Theorem 4. *For* $a \neq 0$*, there is a Bayesian Nash Equilibrium for the second-price auction with regular friendship graphs in which all bidders bid* $b(v) = \mathrm{E}\left[X \mid X > v\right]$*, where* X *is a random variable with cdf* $1 - (1 - F(x))^{1-\frac{(n-1)\alpha}{ad}}$.

Proof. We again substitute the symmetric guess $b_i = b$ for all i into the system (5), canceling and simplifying it to

$$\begin{aligned} &a_{i,i} \cdot \textstyle\sum_{j\neq i} \frac{f(v)}{F(v)b'(v)} \cdot (v - b(v)) \\ &+ \textstyle\sum_{j\neq i} \frac{a_{i,j}}{F(v)} \cdot \Big(- b(v) \textstyle\sum_{k\neq i,j} \frac{f(v)}{b'(v)} + b(v) \textstyle\sum_{\ell\neq i} \frac{f(v)}{b'(v)} - v\frac{f(v)}{b'(v)} - 1\Big) = - \textstyle\sum_{j\neq i} a_{i,j}. \end{aligned}$$

Noting that the two sums inside the parentheses almost cancel out, pulling constant terms out of the sum, and using that $\sum_{j\neq i} a_{i,j} = da$ and $a_{i,i} = \alpha$ for all i, we simplify further to

$$\alpha \cdot (n-1) \cdot \frac{f(v)}{F(v)b'(v)} \cdot (v - b(v)) - \frac{f(v)}{F(v)b'(v)} \cdot da \cdot (v - b(v)) = -(1 - \frac{1}{F(v)}) \cdot da.$$

Rearranging yields the differential equation $b(v) = v + \frac{-ad\cdot(1-F(v))\cdot b'(v)}{((n-1)\alpha - ad)\cdot f(v)}$, which for $a \neq 0$ has the solution

$$b(v) = \frac{1}{(1-F(v))^{1-\frac{(n-1)\alpha}{ad}}} \cdot \int_v^1 x \cdot (1 - \frac{(n-1)\alpha}{ad}) \cdot (1 - F(x))^{-\frac{(n-1)\alpha}{ad}} f(x) dx.$$

Thus, we have proved the theorem. ■

(Note that for $a = 0$, the differential equation simplifies to $b(v) = v$, which matches the known truthful bidding strategy for standard second-price auctions.) The bidding function can be interpreted as the expectation of the lowest of $1 - \frac{(n-1)\alpha}{ad}$ private values above v. Substituting the uniform distribution over $[0,1]$ for F gives us the following corollary:

Corollary 3. *There is a symmetric Bayesian Nash Equilibrium for the second-price auction with all bids independently and uniformly drawn from* $[0,1]$ *in which all bidders bid*

$$b(v) = (1 + \tfrac{ad}{(n-1)\alpha - 2ad}) \cdot v - \tfrac{ad}{(n-1)\alpha - 2ad}.$$

Again, when $d = n-1$, Theorem 4 and Corollary 3 subsume the results for second-price auctions with uniform spite by Brandt et al. [6] who showed that $b(v) = \frac{v-a}{1-a}$ (with $\alpha = 1 + a$), and those by Morgan et al. [18] (with $\alpha = 1$). By combining Corollaries 2 and 3, we can compare the expected revenues of the first-price auction and second-price auction when all valuations are drawn from the uniform distribution.

Theorem 5. *Assume that the social graph is regular, with uniform spite/friendship values* $a < \alpha$*, and that the valuations of all bidders are drawn independently and uniformly from* $[0,1]$*. Then,*

1. *In the presence of uniform spite* ($a < 0$)*, the expected revenue of the second-price auction dominates the expected revenue of the first-price auction.*
2. *In the presence of uniform altruism* ($a > 0$)*, the expected revenue of the first-price auction dominates the expected revenue of the second-price auction.*

Proof. Let b_F and b_S denote the bidding functions for first- and second-price auctions, respectively. Also, let $V_{(1)}$ and $V_{(2)}$ be the highest and second-highest valuations among all bidders, respectively. Notice that because all bidders use the same bidding function, the highest valuation always corresponds to the highest bid, and the second-highest valuation to the second-highest bid.

The revenue of the first-price auction is thus $b_F(V_{(1)})$, while the revenue of the second-price auction is $b_S(V_{(2)})$. Notice that both bidding functions are linear, so we can use linearity of expectations. Furthermore, $\mathrm{E}\left[V_{(1)}\right] = \frac{n}{n+1}$, and $\mathrm{E}\left[V_{(2)}\right] = \frac{n-1}{n+1}$. Substituting these in the bidding functions of Corollaries 2 and 3,

$$\begin{aligned}
\mathrm{E}\left[b_F(V_{(1)})\right] &= (1 - \tfrac{\alpha}{n\cdot\alpha - ad}) \cdot \tfrac{n}{n+1} = \tfrac{(n-1)\alpha - ad}{n\cdot\alpha - ad} \cdot \tfrac{n}{n+1},\\
\mathrm{E}\left[b_S(V_{(2)})\right] &= (1 + \tfrac{ad}{(n-1)\alpha - 2ad}) \cdot \tfrac{n-1}{n+1} - \tfrac{ad}{(n-1)\alpha - 2ad}\\
&= \tfrac{(n-1)\alpha - ad}{(n-1)\alpha - 2ad} \cdot \tfrac{n-1}{n+1} - \tfrac{ad}{(n-1)\alpha - 2ad}.
\end{aligned}$$

The difference is

$$\begin{aligned}
&\mathrm{E}\left[b_S(V_{(2)})\right] - \mathrm{E}\left[b_F(V_{(1)})\right] = b_S(\mathrm{E}\left[V_{(2)}\right]) - b_F(\mathrm{E}\left[V_{(1)}\right])\\
&= \tfrac{(n-1)\alpha - ad}{(n-1)\alpha - 2ad} \cdot \tfrac{n-1}{n+1} - \tfrac{ad}{(n-1)\alpha - 2ad} - \tfrac{(n-1)\alpha - ad}{n\cdot\alpha - ad} \cdot \tfrac{n}{n+1}\\
&= \tfrac{(n-1)\cdot(n\cdot\alpha - ad)((n-1)\alpha - ad) - n\cdot((n-1)\alpha - 2ad)\cdot((n-1)\alpha - ad)}{(n+1)\cdot((n-1)\alpha - 2ad)\cdot(n\cdot\alpha - ad)} - \tfrac{ad}{(n-1)\alpha - 2ad}\\
&= \tfrac{-ad\cdot(n\cdot\alpha - ad)}{((n-1)\alpha - 2ad)\cdot(n\cdot\alpha - ad)} - \tfrac{-\alpha(n-1)ad + (ad)^2}{((n-1)\alpha - 2ad)\cdot(n\cdot\alpha - ad)}\\
&= \tfrac{-ad\cdot\alpha}{((n-1)\cdot\alpha - 2ad)\cdot(n\cdot\alpha - ad)}.
\end{aligned}$$

Because $\alpha > 0$ by definition, the denominator is positive for all values of d and all $a \in (-1, 1)$. The numerator has the opposite sign of a. Thus, we have proved the claim. ■

Notice that Theorem 5 recovers a special case of Theorem 3 from [6] for $\alpha = 1 + a$ and $a < 0$. However, [6] proved the result for arbitrary valuation distributions, while ours holds only for uniform valuations. The techniques used in [6] do not carry over immediately when the degree d of agents is small, and generalizing Theorem 5 to arbitrary distributions is ongoing work.

4 Conclusions

In this paper, we studied auctions with spite and altruism among bidders. We gave explicit characterizations of Nash Equilibria for first-price auctions with valuations drawn uniformly from $[0, 1]$ and arbitrary spite/altruism matrices A, and for first- and second-price auctions with arbitrary valuations and regular social networks.

Many questions remain for future work. For Bayesian auctions, can we find a Nash Equilibrium for second-price auctions in general? It appears that this is significantly more complex: the fact that first-price auctions had a Nash Equilibrium in which each bidder simply multiplies her bid by a constant was fortuitous. Also, can we extend the analysis of first-price auctions to other distributions, or to priors that are not identical for different bidders? Even within the realm we considered, it would be interesting to characterize *all* Nash Equilibria, although this has proven to be quite difficult even in simpler settings.

Having characterized the Nash Equilibrium bidding strategies, we would also like to explicitly compute the revenue and social welfare of the auction. The main obstacle here is to find the expected value of the winning bid, which is now a maximum among n values drawn from different distributions. A secondary problem is that the range of each distribution ($[0, \gamma_i]$) is only given by a formula involving determinants. Calculating the revenue or social welfare would let us characterize a "price of spite" or "benefit of altruism".

Another intriguing question is whether agents can learn equilibrium bidding strategies using a natural algorithm. Assuming that each agent knows the entire matrix A is certainly unrealistic. Are there simple strategies (in the style of [5]) wherein each bidder adapts her bidding strategy based on the utility derived from earlier auctions?

Finally, we would like to extend these results beyond single-item auctions to more complex settings. A particularly promising direction would be the context of keyword auctions [15], as well as various combinatorial settings.

Acknowledgments. We would like to thank Felix Brandt, Harrison Cheng, and Yoav Shoham for useful discussions and references. We also would like to thank several anonymous referees for useful feedback.

References

1. Ashlagi, I., Krysta, P., Tennenholtz, M.: Social context games. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 675–683. Springer, Heidelberg (2008)

2. Aspnes, J., Chang, K., Yamposlkiy, A.: Inoculation strategies for victims of viruses and the sum-of-squares partition problem. Journal of Computer and System Sciences 72(6), 1077–1093 (2006)
3. Babaioff, M., Kleinberg, R., Papadimitriou, C.: Congestion games with malicious players. In: Proc. 9th ACM Conf. on Electronic Commerce, pp. 103–112 (2007)
4. Bergstrom, T.: Systems of benevolent utility functions. J. of Public Economic Theory 1(1), 71–100 (1999)
5. Blum, A., Even-Dar, E., Ligett, K.: Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games. In: Proc. 25th ACM Symp. on Principles of Distributed Computing, pp. 45–52 (2006)
6. Brandt, F., Sandholm, T., Shoham, Y.: Spiteful bidding in sealed-bid auctions. In: Proc. 22nd Intl. Joint Conf. on Artificial Intelligence, pp. 1207–1214 (2007)
7. Brandt, F., Weiß, G.: Antisocial agents and Vickrey auctions. In: Proc. 8th Workshop on Agent Theories, Architectures and Languages, pp. 335–347 (2001)
8. Brocas, I.: Endogenous entry in auctions with negative externalities. Theory and Decision 54(2), 125–149 (2003)
9. Chen, P., Kempe, D.: Altruism and selfishness in traffic routing. In: Proc. 10th ACM Conf. on Electronic Commerce, pp. 140–149 (2008)
10. Deng, X., Qi, Q.: Priority right auction for komi setting (2009), Available at SSRN, http://ssrn.com/abstract=1405294
11. Jehiel, P., Moldovanu, B.: Allocative and informational externalities in auctions and related mechanisms. In: Proc. 9th World Congress of the Econometric Society (2006)
12. Jehiel, P., Moldovanu, B., Stacchetti, E.: How (not) to sell nuclear weapons. American Economic Review 86(4), 814–829 (1996)
13. Jehiel, P., Moldovanu, B., Stacchetti, E.: Multidimensional mechanism design for auctions with externalities. Journal of Economic Theory 85(2), 258–294 (1999)
14. Ledyard, J.: Public goods: A survey of experimental resesarch. In: Kagel, J., Roth, A. (eds.) Handbook of Experimental Economics, pp. 111–194. Princeton University Press, Princeton (1997)
15. Liang, L., Qi, Q.: Cooperative or vindictive: Bidding strategies in sponsored search auctions. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 167–178. Springer, Heidelberg (2007)
16. Maasland, E., Onderstal, S.: Auctions with financial externalities. Economic Theory 32(3), 551–574 (2007)
17. Meier, D., Oswald, Y., Schmid, S., Wattenhofer, R.: On the windfall of friendship: Inoculation strategies on social networks. In: Proc. 10th ACM Conf. on Electronic Commerce, pp. 294–301 (2008)
18. Morgan, J., Steiglitz, K., Reis, G.: The spite motive and equilibrium behavior in auctions. Contributions to Economic Analysis & Policy 2(1), 1102–1127 (2003)
19. Moscibroda, T., Schmid, S., Wattenhofer, R.: When selfish meets evil: Byzantine players in a virus inoculation game. In: Proc. 25th ACM Symp. on Principles of Distributed Computing, pp. 35–44 (2006)
20. Roth, A.: The price of malice in linear congestion games. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 118–125. Springer, Heidelberg (2008)
21. Vetsikas, I., Jennings, N.: Outperforming the competition in multi-unit sealed bid auctions. In: Proc. 6thIntl. Conf. on Autonomous Agents and Multiagent Systems, pp. 702–709 (2007)

On Equilibria for ADM Minimization Games

Leah Epstein[1] and Asaf Levin[2]

[1] Department of Mathematics, University of Haifa, 31905 Haifa, Israel
lea@math.haifa.ac.il
[2] Chaya Fellow. Faculty of Industrial Engineering and Management, The Technion, 32000 Haifa, Israel
levinas@ie.technion.ac.il

Abstract. In the ADM minimization problem, the input is a set of arcs along a directed ring. The input arcs need to be partitioned into non-overlapping chains and cycles so as to minimize the total number of endpoints, where a k-arc cycle contributes k endpoints and a k-arc chain contains $k+1$ endpoints. We study ADM minimization problem both as a non-cooperative and a cooperative games. In these games, each arc corresponds to a player, and the players share the cost of the ADM switches. We consider two cost allocation models, a model which was considered by Flammini et al., and a new cost allocation model, which is inspired by congestion games. We compare the price of anarchy and price of stability in the two cost allocation models, as well as the strong price of anarchy and the strong price of stability.

1 Introduction

WDM (Wavelength Division Multiplexing)/SONET (Synchronous Optical NETworks) rings form a network architecture, which is being used, for example, by telecom carriers. In such architectures, each wavelength channel carries a high-speed SONET ring. The key terminating equipment consists of optical add-drop multiplexers (OADM) and SONET add-drop multiplexers (ADM). Each vertex is equipped with exactly one OADM. For a given ring, a SONET ADM is required at every vertex which carries some traffic terminating at this vertex, but not at other vertices. This motivates problems where the costs incurred by SONET ADMs are considered, with the goal of minimizing such costs.

Formally, we are given a set E of circular-arcs over the vertices $0, 1, \ldots, n-1$, where the vertices are ordered clockwise. The vertices $0, 1, \ldots, n-1$ form a ring, where a link connects vertex $i-1$ to vertex i, for $i = 1, 2, \ldots, n-1$, and in addition, there is a link connecting vertex $n-1$ to vertex 0. An arc (i, j) represents the clockwise path along the ring from a vertex i to the vertex j. Each arc is operated by a selfish rational player. A pair of arcs $(i, j), (k, l)$ is *non-intersecting* if the clockwise path along the cycle $0, 1, \ldots, n-1, 0$ that connects i to j and the clockwise path that connects k to l do not share any link of the cycle. A set of arcs is non-intersecting if each pair of arcs from this set is non-intersecting. A feasible solution is a partition of E into non-intersecting

M. Mavronicolas and V.G. Papadopoulou (Eds.): SAGT 2009, LNCS 5814, pp. 347–358, 2009.

subsets of arcs $E_1, E_2, \ldots, E_p$. The social cost of E_i is the number of different vertices of the ring which are end-points of any arc in E_i. The social cost of the solution is the sum of social costs of E_i for all i. The social goal is to find a minimum social cost feasible solution.

For an arc (i, j), we define its *length* as $\ell(i, j) = j - i \mod n$. For a subset of arcs, the length of the subset is the total length of its arcs. Throughout the paper, the length of a path is defined to be the total length of its arcs. We use the following auxiliary definition. *The deficiency of a vertex* v, $def(v)$, is defined as follows. Let $in(v)$ be the number of ingoing arcs of v, and let $out(v)$ be the number of outgoing arcs of V. Then, $def(v) = \frac{1}{2}|in(v) - out(v)|$. The deficiency of a set of arcs is the total deficiency over all vertices.

A *chain* is an open directed path of length at most $n-1$, and a *cycle* is a closed directed path of length exactly n. Without loss of generality, we can assume that the arcs in each E_i form a connected component (either a chain or a cycle). This is so because if the arcs in E_i are disconnected, then we can partition E_i to its connected components without increasing its total social cost. Therefore, we ask for a partition of E into cycles and (open-)chains. Since the social cost of a cycle equals to the number of arcs in it, whereas the social cost of a chain equals to the number of arcs in it plus 1, an alternative definition for the social cost of a feasible solution is given by the sum of the two factors, the *number of chains* in the solution and the *number of arcs* in the input, $|E|$.

A significant amount of work on the minimization problem of the social goal was carried out during the last few years. The current best approximation ratio for this global problem is a $\frac{98}{69} \approx 1.42029$-approximation algorithm in [6]. For additional work on the problem, see [21,6,9,14,10,22,4].

In this paper we are concerned with the selfish behavior of users corresponding to the arcs of the above problem. Flammini et al. [8] considered two non-cooperative games related to this problem. In their setting the cost of a solution is distributed among the arcs, and each arc tries selfishly to improve its own cost by changing its connections (to its adjacent arcs), without disconnecting any existing connections in which it does not participate. In these games they analyzed the so-called *price of anarchy* (see below). The two games of [8] differ by the cost allocation schemes (i.e., how one distributes the cost of the solution among the arcs). They considered two allocation schemes. In the first one, called Shapley, the arcs sharing an ADM pay for it by splitting its cost equally. That is, an arc is charged with two costs resulting from its endpoints as follows. If two arcs appear consecutively along a common chain or cycle then the single ADM which is required at this common endpoint implies identical costs of $\frac{1}{2}$, charged to each one of the two arcs. An arc which has an endpoint which is not shared with another arc is charged with a unit cost, which results from the ADM required at this endpoint. In the Egaliterian cost allocation scheme the cost of the entire solution is shared equally by all arcs. Hence in the Egaliterian method each player basically has the social goal as its own goal. They showed that the price of anarchy in each of the two models is exactly $\frac{5}{3}$. The results of [8] hold also for general topology but they were in fact shown to be tight already for rings, which we consider here. They concluded their work by stating that "the determination of other intermediate methods combining both the Shapley and Egaliterian advantages is an important left open question". A general discussion on cost allocation methods for combinatorial optimization problems can be found in [11]. Note that a meaningful cost sharing method for the ADM problem should be such that the cost of a player is related to the connections of its arc on one hand, similarly to the Shapley model, but as in the Egaliterian model, the cost cannot be determined in an entirely local way. Not only direct connections should affect the cost, but indirect connections should be taken into account as well. In this paper we introduce such a cost allocation model and study its properties.

In the game definition of [8], each arc (i.e., player) has a strategy defined as a pair consisting of its clockwise adjacent arc (if such exists), and its counterclockwise adjacent arc. We note that given a solution, one arc cannot change its strategy without affecting the strategies of other arcs, and hence we must allow an arc to affect the strategies of other arcs in a limited way as follows. An arc a can force a change of the strategies of its current adjacent arcs at their common endpoints by disconnecting a's existing neighbors. In addition, it can force the new adjacent arcs of a to connect to a, but only if the endpoints which are being connected to a were free before the change (assuming such a change creates a feasible solution, i.e., in the new solution the component which contains a does not have intersecting arcs). More precisely, a new adjacent arc b can be connected to a only if in the previous solution the relevant endpoint of b was not connected at all. That is, an arc a can force a pair of connected arcs b, c to become disconnected, only if $a = b$ or $a = c$. An arc a is motivated to deviate from its strategy if as a result of the deviation the cost which it incurs strictly decreases. We define a mapping over the set of valid solutions as follows. One solution is mapped to another solution if the second solution can result from the first one by a deviation of a single arc from its strategy (possibly forcing the change of strategies of its previously adjacent arcs, and new adjacent arcs). Each fixed point of the mapping is considered as a (pure) Nash equilibrium (an NE) [17]. If we consider a cooperative game in which we allow a coalition of arcs to change their strategies simultaneously, then a new solution is created in two steps. In the first step each arc of the coalition can get disconnected from one or two arcs (forcing these arcs a change of strategy). In the second step, each arc of the coalition can connect to other arcs at its free endpoints, provided that no additional connections are being disconnected, and the implied solution is feasible. An arc would join a coalition if its cost strictly decreases in the new solution. A fixed point of the corresponding mapping defined on the set of solutions is called a strong Nash equilibrium (an SNE), or a strong equilibrium [3].

In this paper we consider the Shapley cost allocation scheme of [8] as well as a new cost allocation scheme which is motivated by congestion games [19]. In congestion games, the total cost of a part of the solution is usually split equally among the players sharing this part (which is typically a resource). In our setting we let the arcs of a common chain or cycle (of the resulting solution) share the cost of their component equally. That is, an arc which belongs to a cycle pays 1, and an arc which belongs to a chain consisting of k arcs pays $1 + \frac{1}{k}$. We call this cost allocation, Congestion cost allocation.

For the two models, we present upper and lower bounds on the price of anarchy, the strong price of anarchy, price of stability and strong price of stability. We next define these metrics of games. For a game G which belongs to a model M, which admits an NE and an SNE, we let $\mathrm{OPT}(G)$ denote the minimum social cost of any solution (when G is clear from the context, we denote this optimal solution and its cost by OPT). Let $\mathrm{NE}(G)$ denote the set of valid solutions which are Nash equilibria and let $\mathrm{SNE}(G)$ denote the set of valid solutions which are strong equilibria. For a solution S, we use S to denote its social cost as well. Then the *price of anarchy* for the model M is defined as $\mathrm{PoA}(M) = \sup_{G \in M} \sup_{S \in \mathrm{NE}(G)} \frac{S}{\mathrm{OPT}(G)}$, that is, the worst case ratio between the social cost of a solution which is a Nash equilibrium, and the optimal social cost. The *strong price of anarchy* is defined as $\mathrm{SPoA}(M) = \sup_{G \in M} \sup_{S \in \mathrm{SNE}(G)} \frac{S}{\mathrm{OPT}(G)}$, that is, the worst case ratio between the largest social cost of a solution which is a strong equilibrium, and the optimal social cost. The *price of stability* and the *strong price of stability* are defined as $\mathrm{PoS}(M) = \sup_{G \in M} \inf_{S \in \mathrm{NE}(G)} \frac{S}{\mathrm{OPT}(G)}$ and $\mathrm{SPoS}(M) = \sup_{G \in M} \inf_{S \in \mathrm{SNE}(G)} \frac{S}{\mathrm{OPT}(G)}$, respectively, that is, the worst case ratio between the smallest social cost of a solution which is a strong equilibrium, and the optimal social cost.

Since for any game G, $\mathrm{SNE}(G) \subseteq \mathrm{NE}(G)$, if all the measures above are well defined, a model M satisfies $\mathrm{PoS}(M) \leq \mathrm{SPoS}(M) \leq \mathrm{SPoA}(M) \leq \mathrm{PoA}(M)$.

In the last few years, there has been extensive studies of the relation between equilibria for combinatorial optimization problems, and social optima. The application of concepts and techniques borrowed from game theory to various problems in computer science, and specifically, to network problems, was initiated in [13,18]. Since then, issues like *routing* [20,15,5], *bandwidth allocation* [23], and *congestion control* [12], to name only a few, have been analyzed from a game theoretic perspective. Typically the studies focus on the worst equilibrium of a given model (that is, on the price of anarchy), since the price of stability often turns out to be 1, yet systems may converge to any possible equilibrium. Since for some applications, a system can be forced to stay in a specific configuration, as long as this configuration is stable, the best equilibrium is of interest too, and a more careful study of the price of stability is required [2]. There has been some work on cooperative games as well, and in particular, Andelman, Feldman, and Mansour [1] suggested the study of strong equilibria in order to separate the effect of lack of coordination from the effect of selfishness.

The Egaliterian cost sharing model for the ADM minimization problem was shown to be equivalent to the Shapley cost sharing method, in terms of Nash equilibria [8]. This property is not true for strong equilibria. It is not difficult to see that SPoA(Egaliterian)= 1, and therefore PoS(Egaliterian) = SPoS(Egaliterian) = 1. This holds since a solution which is not a social optimum cannot be a strong equilibrium; for such a solution, the entire set of arcs is a coalition which wishes to deviate and create a social optimum. Thus we do not consider this variant further in the current work. Interestingly, we show that SPoA(Shapley) $= \frac{3}{2}$, but yet SPoS(Shapley) $= 1$. The Congestion cost sharing model turns out to be similar to the other models in terms of the PoA, and we show PoA(Congestion) $= \frac{5}{3}$. We further show $\frac{3}{2} \leq$ SPoA(Congestion) $\leq \frac{11}{7} \approx 1.57143$. However, in this model the best equilibria turns out to be of interest. We show PoS(Congestion) $\geq \frac{16}{13} \approx 1.23077$, and $1.25 \leq$ SPoS(Congestion) ≤ 1.38. The upper bounds on the SPoA and the SPoS of the Congestion model are our main technical contribution. In order to obtain these bounds, we present insights into the combinatorial structure of the problem. We then use factor revealing mathematical programs and solve them using a non-trivial application of methods allowing to solve rational goal optimization problems with linear constraints.

In the full version, we discuss an additional ADM minimization problem, called the *chord version* [4,7]. Due to space limitation, some proofs are omitted.

Table 1. Results. The abbreviations LB and UB stand for lower bound and upper bound, respectively. The abbreviations E., S. and C. stand for the cost allocation models: Egaliterian, Shapley and Congestion, respectively.

	E. LB	E. UB	S. LB	S. UB	C. LB	C. UB
PoA	5/3 [8]	5/3 [8]	5/3 [8]	5/3 [8]	5/3	5/3
SPoA	1	1	3/2	3/2	3/2	$11/7 \approx 1.57143$
SPoS	1	1	1	1	1.25	1.38
PoS	1 [8]	1 [8]	1 [8]	1 [8]	$16/13 \approx 1.23077$	1.38

2 The Shapley Cost Allocation Model

In this section, we consider the Shapley cost allocation model. The paper [8] proved that the price of anarchy of this game is exactly $\frac{5}{3}$. We show that the corresponding game always admits an SNE, and consider the strong price of anarchy of this game, which turns out to be very different from the SPoA in the Egaliterian cost sharing model.

Theorem 1. *An optimal solution* OPT*, which has a maximal number of cycles (among optimal solutions), is an* SNE*. Therefore, an* SNE *always exists, and* PoS = SPoS = 1*. The strong price of anarchy of the* Shapley *cost allocation model is exactly* $\frac{3}{2}$*.*

3 The Congestion Cost Allocation Model

In this section we study the new cost allocation model and stress the differences between the two models. Since the arcs which belong to a chain share its cost equally, arcs are well motivated to belong to long chains. We start with showing two simple games, each of which corresponds to a Nash equilibrium in one model but not in the other. By these examples, we find that the two models are distinct, and the new model needs to be considered separately.

Example 1. Consider a ring over $n = 5$ vertices, and the four arcs $e_1 = (0, 2)$, $e_2 = (2, 3)$, $e_3 = (3, 4)$ and $e_4 = (4, 1)$. The total length of arcs is 6, and moreover the arcs do not induce a valid cycle, therefore any solution consists of at least two chains. Consider the solution which consists of two chains, where the first chain contains e_1 and e_2 and the second one contains the other two arcs. This solution is an SNE in the Shapley model; even though either e_2 or e_3 could potentially connect to the other chain, since the cost of each arc is $\frac{3}{2}$, this cost would not change as a result of the deviation. However, this deviation is beneficial in the Congestion model, where this solution is not an NE, since it would change the cost of a deviating arc from $\frac{3}{2}$ to $\frac{4}{3}$.

The first example showed a game which is an SNE in the Shapley model, but not even an NE in the Congestion model. This situation seems to be natural, since costs are allocated in the Shapley model only as a function of the role of an arc in a chain or a cycle, and in a more uniform way in the Congestion model. We show however, that the opposite situation may occur as well.

Example 2. Consider a ring over $n = 6$ vertices, and the five arcs $e_1 = (0, 1)$, $e_2 = (1, 3)$, $e_3 = (3, 5)$, $e_4 = (4, 0)$ and $e_5 = (1, 2)$. The total length of arcs is 8, and the arcs do not induce a valid cycle, therefore any solution consists of at least two chains. Consider the solution which consists of three chains, where the first chain contains e_1, e_2, and e_3, and each of e_4 and e_5 are one-arc chains. This solution is an SNE in the Congestion model; the longest chain which can be created from the input consists of three arcs, so the arcs of the first chain have no incentive to change their strategies. However, the deviation of e_1, to join both e_4 and e_5 is beneficial in the Shapley model, since its cost would change from $\frac{3}{2}$ to 1, so this solution is not even an NE in the Shapley model.

A crucial difference between the two cost allocation models is that in the Congestion model the costs of arcs can take any value of the form $1 + \frac{1}{k}$ for $k = 1, 2, \ldots, \infty$, while the costs in the Shapley model are in the set $\{1, \frac{3}{2}, 2\}$. Still, an arc which belongs to a cycle pays 1, and thus it would never deviate from its current strategy.

We next establish the existence of an SNE for any game of this model, which will ensure the existence of an NE in every game as well. To prove the existence of an SNE we study the following algorithm.

Algorithm A

1. Repeat until no valid cycle can be created:
   Find a valid cycle and remove its arcs from the input. Add the newly found cycle to the components of the resulting solution.
2. Repeat until the input is empty:
   Find a valid chain with a maximum number of arcs and remove it from

the input. Add the newly found chain to the components of the resulting solution.

end of algorithm A.

The relation between this algorithm A and strong equilibria is established in the next lemma.

Lemma 1. *Algorithm A is a polynomial time algorithm which always returns an* SNE*. Moreover, every* SNE *is an output of some execution of algorithm A (i.e., can be returned by the algorithm with some tie-breaking decisions).*

Theorem 2. *For any game in the* Congestion *cost allocation model there exists an* SNE *(and therefore an* NE*). It is possible to verify in polynomial time that a solution* SOL *is an* SNE*.*

Proof. The existence of an NE and an SNE follows by Lemma 1. It remains to show a polynomial time algorithm which validates that some solution SOL is an SNE. We consider the set of arcs which belong to chains in SOL. If these arcs contains the arcs of a valid cycle, then SOL is clearly not an SNE. Otherwise, for each $i = 1, 2, \ldots, n-1$ we check if the collection of arcs which belong to chains of at most i arcs in SOL can form a chain consisting of at least $i+1$ arcs (by finding a valid chain with a maximum number of arcs). If so, then SOL is not an SNE. Otherwise SOL is indeed an SNE, since it can be given as an output of algorithm A. □

After we have established the existence of an NE, we analyze the price of anarchy of this game. Note that the proof of [8] is not valid for this case, since it relies on the fact that an arc has one of a set of three possible costs.

Theorem 3. *The price of anarchy of the* Congestion *cost allocation game is exactly $\frac{5}{3}$. The strong price of stability of the* Congestion *cost allocation model is at least $\frac{5}{4} = 1.25$. The price of stability in the* Congestion *cost allocation model is at least $\frac{16}{13}$. The strong price of anarchy of the* Congestion *cost allocation model is at least $\frac{3}{2}$.*

Theorem 4. *The* SPoS *of the* Congestion *cost allocation model is at most* 1.38*.*

Proof. The overview of the proof is as follows. We obtain combinatorial structural properties of a pre-specified SNE, and use these properties to get a mathematical program which bounds the average cost of the SNE per a unit cost of the optimal solution. The calculation is performed on all components of OPT together. This mathematical program has a ratio objective function, and linear constraints. We use the properties of this type of programs to obtain a closed-form solution, which enables us to prove the claim.

Consider a given input consisting of a set of m arcs $|E|$ on a ring with n vertices and an optimal solution OPT. We construct an SNE, SOL, using a specific execution of the greedy algorithm above. In the cycle creation phase, at every step, if there exists a cycle of OPT for which all arcs are still available, then such a cycle is chosen. In the chain creation phase, let i be the maximum number of arcs in any valid chain which can be created from available arcs. If there exists a chain or a sub-path of a chain of OPT, which consists of i arcs, and all these arcs are still available, then such a chain is chosen.

Without loss of generality, we assume that the two solutions OPT and SOL do not contain any identical component (cycle or chain), and in particular, no common cycles, and no common chains consisting of a single arc. If such a common component exists, the removal of its arcs from the input would not harm the strong equilibrium, and would not increase the ratio between the costs of SOL and the cost of OPT. By the construction, this means that OPT consists of chains and no cycles, since all its cycles are part of the constructed solution.

We create a new graph over the vertices of the ring as follows. Let $C = \{p_1, p_2, \ldots, p_i\}$ be a chain of SOL, such that $p_j = (\ell_j, r_j)$. Then the arcs of C are removed and replaced by a single arc (ℓ_1, r_i). The arcs which are inserted instead of the original arcs are called the *new arcs*. Note that the deficiency of the new graph is equal to the deficiency of the original set of arcs, and the total length of new arcs is equal to the total length of original arcs.

We claim that if two new arcs share an endpoint in the new graph, then the corresponding two chains have a total length which exceeds n. To see this, we note that otherwise the arcs of the two chains can form a coalition which would reduce the cost of every arc in it by creating a concatenated chain. Denote the new set of new arcs by E'. Let $d = def(E) = def(E')$. We claim that the new arcs can be partitioned into (possibly invalid) paths and cycles, such that the number of paths is at most d. This can be achieved by constructing Euler tours on the connected components of the new graph. We next claim that the total length of new arcs is at least $n\frac{|E'|-d}{2}$. Consider a cycle of k new arcs in the partition into cycles and paths which is implied by the Euler decomposition. Since the length of each pair of consecutive new arcs is more than n, we get that the total length of the cycle is at least $\frac{kn}{2}$. A similar argument holds for paths with an even number of new arcs, since the path can be split into $\frac{k}{2}$ consecutive pairs. Consider a path of k new arcs, where k is odd. The same argument holds for a sub-path of $k-1$ arcs, and thus the total length is at least $\frac{(k-1)n}{2}$. Summing up for all components, and taking into account that the number of odd paths is at most the number of paths which is at most d, we get a total length of $n\frac{|E'|-d}{2}$.

We next find lower bounds on the number of chains in OPT. Since OPT contains no cycles, and the length of each chain is at most $n-1$, then the number of chains is at least the total length of all arcs divided by $n-1$. Since the total length of arcs in E is equal to the total length of arcs in E', OPT contains more than $\frac{|E'|-d}{2}$ chains. On the other hand, OPT has at least d chains due to the deficiency. Therefore, $|E'| = 2(\frac{|E'|-d}{2}) + d$ is at most three times the number of chains of OPT.

Denote by CH_i the number of chains with i arcs in OPT. Then we derive that the number X of chains in SOL satisfies $X \leq 3 \sum_{i=1}^{n-1} CH_i$. We next obtain another upper bound on X. To do so, we will bound the total price paid by all arcs. We denote by $T(i) + i$ an upper bound on the maximum total price which can be paid by the arcs of one chain, which consists of i arcs, that is, $T(i)$ is the marginal cost of the arcs, not taking into account the cost of at least 1, which is the minimal cost for any arc.

Our bound on $T(i)$ will follow from the following observation. Given an i-arcs chain C, it must contain an arc which pays at most $\frac{1}{i+1}$ (otherwise, in the creation process of SOL, we would prefer to pick C which is part of the optimal solution). When we delete this arc from a chain we create two chains with j arcs and $i - j - 1$ arcs and the total price paid by the arcs of these chains is at most $T(j) + T(i - j - 1)$. Hence, $T(i)$ satisfies the following recursion.

$$T(i) = \frac{1}{i+1} + \max_{j=0,1,2,\ldots,\lfloor (i-1)/2 \rfloor} \{T(j) + T(i - j - 1)\}.$$

The starting condition is $T(0) = 0$ and $T(1) = 1$. Via direct calculation we can find the following values which we use later : $T(2) = \frac{4}{3}$, $T(3) = \frac{9}{4} = 2.25$, $T(4) = \frac{38}{15} \approx 2.5333$, $T(5) = \frac{41}{12} \approx 3.41667$, $T(6) = \frac{313}{84} \approx 3.72619$, $T(7) = \frac{37}{8} = 4.625$. Moreover, since each chain incurs a cost of 1 in addition to a cost of 1 for each one of its arcs, we conclude that $X \leq \sum_{i=1}^{n-1} T(i)CH_i$.

We next formulate a mathematical program whose solution is an upper bound on the SPoS. We let $x = \frac{X}{m}$ and for all i, $c_i = \frac{CH_i}{m}$. We use $\sum_{i=1}^{n-1} iCH_i = m$, $\text{OPT} = m + \sum_{i=1}^{n-1} CH_i$, and $\text{SOL} = m + X$.

$$\max \frac{1 + x}{1 + \sum_{i=1}^{n-1} c_i}$$

$$\textbf{s.t.} \quad \sum_{i=1}^{n-1} ic_i = 1, \; x \leq \sum_{i=1}^{n-1} T(i)c_i, \quad x \leq 3\sum_{i=1}^{n-1} c_i, \quad x, c_1, \ldots, c_{n-1} \geq 0.$$

We next note that since we assume that OPT and SOL do not contain any identical component, and in particular, no common chains consisting of a single arc. Hence in this mathematical program we can replace $T(1)$ by $\frac{1}{2}$ (while the recursive definition still uses $T(1) = 1$).

We next note that though the resulting mathematical program is not a linear program (as the objective function is rational function and non-linear), the optimal solution can be assumed to be a basic solution. To derive this property, we suggest to invoke Megiddo's parametric search method [16] for solving this mathematical program. In this method, we solve the problem by calling an LP solver multiple times. Eventually, the solution is obtained as the solution returned by the LP solver in one of the iterations. If in each iteration we return a basic solution, the overall optimal solution which we find, is a basic solution.

Hence, since there are three constraints (beside the non-negativity constraints) there are at most three positive variables in the optimal solution.

We next note that in the optimal basic solution x must be positive, and hence x is in the optimal basis. It remains to identify the two remaining basic variables. We first show that neither of the surplus variables which we introduce to transform the problem into standard form, is in the optimal basis. To see this fact note that in such basic solutions there is at most one variable c_i which is positive. So $x \le T(i)c_i$, if $i > 1$ and $x \le \frac{c_1}{2}$, if $i = 1$. First we consider the cases in which $i \ge 5$. From the first constraint we conclude that $c_i \le \frac{1}{5}$, and hence $\frac{1+x}{1+\sum_{j=1}^{n-1} c_j} \le \frac{1+3c_i}{1+c_i} = 3 - \frac{2}{1+c_i} \le \frac{4}{3}$.

For $i = 1$, $x \le \frac{c_i}{2}$, so we obtain $\frac{1+x}{1+\sum_{j=1}^{n-1} c_j} \le \frac{1+\frac{c_i}{2}}{1+c_i} \le 1$. For $i = 2$, $x \le \frac{4}{3}c_i$, so we obtain $\frac{1+x}{1+\sum_{j=1}^{n-1} c_j} \le \frac{1+\frac{4}{3}c_i}{1+c_i} < \frac{4}{3}$. For $i = 3$, $x \le \frac{9}{4}c_i$, and $c_i \le \frac{1}{3}$, so we obtain $\frac{1+x}{1+\sum_{j=1}^{n-1} c_j} \le \frac{1+\frac{9}{4}c_i}{1+c_i} = \frac{9}{4} - \frac{5}{4(1+c_i)} \le 1.3125$. For $i = 4$, $x \le \frac{38}{15}c_i$, and $c_i \le \frac{1}{4}$, so we obtain $\frac{1+x}{1+\sum_{j=1}^{n-1} c_j} \le \frac{1+\frac{38}{15}c_i}{1+c_i} \le \frac{38}{15} - \frac{23}{15(1+c_i)} \le \frac{98}{75} \approx 1.30667$.

Therefore, without loss of generality we can restrict ourselves to basic solutions in which all the three constraints are tight. That is, we replace all the inequalities by equalities.

We note that in this optimization problem the objective function, $3 - \frac{6}{x+3}$, is a monotonically increasing function of x. Hence, to optimize it, it suffices to maximize x instead of maximizing $3 - \frac{6}{x+3}$.

We suggest a feasible solution to the linear program. In this solution the basic variables are x, c_3, c_7 and the value of these variables are $x = \frac{19}{27}$, $c_3 = \frac{13}{81}$ and $c_7 = \frac{2}{27}$ (all remaining variables are 0). This solution is clearly a feasible solution to the linear program. We next argue that this is an optimal solution. To prove this claim we consider the dual problem to the above linear program which we state next. The variables α, β and γ correspond to the constraints of the primal program (in this order).

$$\min \alpha, \ \textbf{s.t. } \beta + \gamma \ge 1, \ \alpha - \frac{\beta}{2} - 3\gamma \ge 0, \ j\alpha - T(j)\beta - 3\gamma \ge 0, \ 2 \le j \le n-1.$$

For this dual linear program we present the following dual solution $\alpha = \frac{19}{27}$, $\beta = \frac{32}{27}$ and $\gamma = \frac{-5}{27}$. Note that the objective function value of this dual solution is exactly the objective function value of the primal solution we pinpoint above. Hence, if we establish the feasibility of the dual solution, we get that the primal solution is optimal (by weak duality of linear program).

So we next show that the dual solution is feasible. $\beta + \gamma = \frac{32}{27} + \frac{-5}{27} = 1$ so the first constraint holds. $\alpha - \frac{\beta}{2} - 3\gamma = \frac{19}{27} - \frac{16}{27} + \frac{15}{27} = \frac{2}{3} > 0$ and so the second constraint holds.

As for the (family of the) third constraint, we prove this constraint by induction. The base cases are $j = 0, 1, 2, \ldots, 6$. We need to prove $19j - 32T(j) + 15 \geq 0$. We have $19 \cdot 0 - 32T(0) + 15 = 15$, $19 \cdot 1 - 32T(1) + 15 = 2$, $19 \cdot 2 - 32T(2) + 15 = \frac{31}{3}$, $19 \cdot 3 - 32T(3) + 15 = 0$, $19 \cdot 4 - 32T(4) + 15 = \frac{149}{15}$, $19 \cdot 5 - 32T(5) + 15 = \frac{2}{3}$, and $19 \cdot 6 - 32T(6) + 15 = \frac{205}{21}$. Next, we prove the inequality for some $J \geq 7$. There exist integer values $j_1, j_2 \geq 0$ such that $J = j_1 + j_2 + 1$ and $T(J) = T(j_1) + T(j_2) + \frac{1}{J+1}$. By the induction hypothesis, $19j_i - 32T(j_i) + 15 \geq 0$ for $i = 1, 2$ thus $19(j_1 + j_2) - 32(T(j_1) + T(j_2)) + 30 \geq 0$. We have $19J - 32T(J) + 15 = 19(j_1 + j_2 + 1) - 32(T(j_1 + T(j_2) + \frac{1}{J+1})) + 15 \geq 19 - \frac{32}{J+1} - 15 = 4(1 - \frac{8}{J+1}) \geq 0$, for $J \geq 7$.

Hence, the dual solution is feasible. As explained above, this means that the primal solution is optimal to the linear program and hence also to our original mathematical program which bounds the SPoS. Finally, by plugging-in the values of our primal solution to the objective function of the mathematical program, we obtain that the SPoS is at most $\frac{69}{50} = 1.38$. □

Note that this analysis is not tight in the sense that the solution of the mathematical program results in a worst case example, where OPT consists only of chains with three arcs and chains with seven arcs, where every second arc has a cost of 2 in SOL. Specifically, the costs of arcs in chains of three arcs are $(2, \frac{5}{4}, 2)$ and the costs of arcs in chains of seven arcs are $(2, \frac{5}{4}, 2, \frac{9}{8}, 2, \frac{5}{4}, 2)$. Hence in SOL all arcs belong to chains of a single arc, chains of four arcs or chains of eight arcs. Consider a chain of SOL which consists of four arcs. Denote these arcs by e_1, e_2, e_3, e_4, and assume that in OPT each e_i appears between the arcs a_i and b_i (so a_i, e_i, b_i is a part of a chain in OPT). All arcs a_i and b_i are single-arc chains in SOL. Then, for $i = 1, 2, 3$ we have that a_{i+1} has a common end-vertex as b_i. Since they are not merged into a single chain in SOL, we conclude that $\ell(a_{i+1}) + \ell(b_i) > n$. However, for every i we have $\ell(a_i) + \ell(e_i) + \ell(b_i) < n$. Hence, we conclude that $\ell(a_1) + \ell(e_1) + \ell(e_2) + \ell(e_3) + \ell(e_4) + \ell(b_4) < n$ and therefore SOL can extend the four arcs chain using a_1 and b_4 contradicting the assumption that SOL is an NE. A similar argument holds for eight-arc chains of SOL as well. Therefore, the exact value of SPoS (Congestion) remains open.

For the SPoA, we can apply a similar proof technique incorporated with application of the structural properties of Megiddo's parametric search method [16], to prove the following upper bound.

Theorem 5. *The* SPoA *of the* Congestion *cost allocation model is at most* $\frac{11}{7} \approx 1.5714$.

References

1. Andelman, N., Feldman, M., Mansour, Y.: Strong price of anarchy. In: Proc. of the 18th Symposium on Discrete Algorithms (SODA 2007), pp. 189–198 (2007)
2. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. In: Proc. of the 45st Annual Symposium on Foundations of Computer Science (FOCS 2004), pp. 295–304 (2004)

3. Aumann, R.J.: Acceptable points in general cooperative n-person games. In: Tucker, A.W., Luce, R.D. (eds.) Contributions to the Theory of Games IV, Annals of Mathematics Study 40, pp. 287–324. Princeton University Press, Princeton (1959)
4. Călinescu, G., Wan, P.-J.: Traffic partition in WDM/SONET rings to minimize SONET ADMs. Journal of Combinatorial Optimization 6, 425–453 (2002)
5. Czumaj, A.: Selfish routing on the internet. In: Leung, J. (ed.) Handbook of Scheduling: Algorithms, Models, and Performance Analysis, ch. 42. CRC Press, Boca Raton (2004)
6. Epstein, L., Levin, A.: Better bounds for minimizing SONET ADMs. Journal of Computer and Systems Sciences (to appear)
7. Epstein, L., Levin, A.: The chord version for SONET ADMs minimization. Theoretical Computer Science 349(3), 337–346 (2005)
8. Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., Zaks, S.: Selfishness, collusion and power of local search for the ADMs minimization problem. Computer Networks 52(9), 1721–1731 (2008)
9. Flammini, M., Shalom, M., Zaks, S.: On minimizing the number of ADMs - Tight bounds for an algorithm without preprocessing. Journal of Parallel and Distributed Computing 67(4), 448–455 (2007)
10. Gerstel, O., Lin, P., Sasaki, G.: Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings. In: Proc. INFOCOM 1998, vol. 1, pp. 94–101 (1998)
11. Jain, K., Mahdian, M.: Cost sharing. In: Nisan, Roughgarden, Tardos, Vazirani (eds.) Algorithmic game theory, pp. 385–410. Cambridge university press, Cambridge
12. Karp, R.M., Koutsoupias, E., Papadimitriou, C.H., Shenker, S.: Optimization problems in congestion control. In: Proc. of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000), pp. 66–74 (2000)
13. Koutsoupias, E., Papadimitriou, C.H.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999)
14. Liu, L., Li, X., Wan, P.-J., Frieder, O.: Wavelength assignment in WDM rings to minimize SONET ADMs. In: Proc. INFOCOM 2000, vol. 2, pp. 1020–1025 (2000)
15. Mavronicolas, M., Spirakis, P.G.: The price of selfish routing. In: Proc of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), pp. 510–519 (2001)
16. Megiddo, N.: Combinatorial optimization with rational objective functions. Mathematics of Operations Research 4(4), 414–424 (1979)
17. Nash, J.: Non-cooperative games. Annals of Mathematics 54(2), 286–295 (1951)
18. Papadimitriou, C.H.: Algorithms, games, and the internet. In: Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), pp. 749–753 (2001)
19. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. International Journal of Game Theory 2(1), 65–67 (1973)
20. Roughgarden, T., Tardos, É.: How bad is selfish routing? In: Proc. of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000), pp. 93–102 (2000)
21. Shalom, M., Zaks, S.: A 10/7+ε approximation for minimizing the number of ADMs in SONET rings. IEEE/ACM Transactions on Networking 15(6), 1593–1602 (2007)
22. Wan, P.-J., Călinescu, G., Liu, L., Frieder, O.: Grooming of arbitrary traffic in SONET/WDM BLSRs. IEEE Journal on Selected Areas in Communications 18, 1995–2003 (2000)
23. Yaïche, H., Mazumdar, R., Rosenberg, C.: A game theoretic framework for bandwidth allocation and pricing in broadband networks. IEEE/ACM Transactions on Networking 8(5), 667–678 (2000)

Author Index